

Universitetet i Oslo  
Institutt for informatikk

# Trafikkoptimering i satellittsystemer

Anders Breivik

Hovedfagsoppgave

Mai 2003





# Forord

Denne hovedfagsoppgaven er skrevet som en del av graden Candidatus Scientiarum i studieretning Anvendt og Industriell Matematikk ved Institutt for Informatikk, Universitetet i Oslo.

Den ferdige oppgaven markerer slutten på et langt og innholdsrikt studium, og det er med udelt glede at jeg her setter punktum og tar fatt på neste fase av livet. Sett gjennom etterpåklokskapens briller har det vært en prosess preget av mottoet “veien er målet”, og omveiene har vært lange og mange. Jeg ville likevel ikke vært noen av opplevelsene foruten, de har formet meg til den jeg er i dag.

Jeg vil gjerne få takke min veileder Geir Dahl for tålmodig veiledning underveis. Jeg vil også takke alle de som ved gjennomlesning og diskusjon har kommet med verdifulle bidrag til oppgavens endelige utforming: Unni, Trude, Jonas, Lars og Lars Preben.

Den største takken går til mine fantastiske foreldre, for deres uforbeholdne støtte og omtanke gjennom studiet! Dette hadde ikke vært mulig uten dere. En varm takk går spesielt til min kjære Unni, for all oppmuntring og kjærighet.

En stor takk går til Jonas, for godt selskap og uvurderlig logistikkhjelp under innspurten. Takk også til hans samboer Aina, for uselvisk utlån av mann og bil. Til slutt vil jeg få takke alle mine gode venner og medstudenter på lesesalen i Forskningsparken. Dere gjorde mine siste år på Universitetet til en fest. **Parken forever!**

Oslo, 15. mai 2003

Anders Breivik



# Innhold

<b>Forord</b>	<b>iii</b>
<b>1 Innledning</b>	<b>1</b>
1.1 Bakgrunn og motivasjon . . . . .	1
1.2 Problem . . . . .	2
1.3 Oppgavens mål . . . . .	3
1.4 Oppgavens struktur . . . . .	3
<b>2 Teori</b>	<b>5</b>
2.1 Optimering . . . . .	5
2.2 Grafteori . . . . .	6
2.2.1 Matchinger . . . . .	6
2.2.2 Litt matriseteori . . . . .	9
2.3 Problem - og algoritmeanalyse . . . . .	9
2.3.1 Kompleksitet . . . . .	10
2.3.2 Heuristikker og tilnærmingsalgoritmer . . . . .	13
<b>3 Problem</b>	<b>17</b>
3.1 Bakgrunn . . . . .	17
3.1.1 FDMA . . . . .	17
3.1.2 CDMA . . . . .	18
3.1.3 TDMA . . . . .	18
3.2 SS/TDMA satellittsystemer . . . . .	18
3.3 SS/TDMA trafikkoptimeringsproblemet . . . . .	19
3.4 Preemption . . . . .	22
<b>4 Matematisk modell</b>	<b>23</b>
4.1 SS/TDMA systemets komponenter . . . . .	23
4.2 Trafikkbehovet . . . . .	23
4.3 TDMA skjemaet . . . . .	25
4.3.1 Matriserepresentasjon . . . . .	25
4.3.2 Grafrepresentasjon . . . . .	27
4.4 Preemption . . . . .	27
4.5 SS/TDMA trafikkoptimeringsproblemet og varianter . . . . .	28

<b>5</b>	<b>Analyse</b>	<b>31</b>
5.1	Skranker og minimumsverdier . . . . .	31
5.1.1	Nedre skranke for $s$ i TOP-s . . . . .	32
5.1.2	Nedre skranke for $s$ i TOP-s-k . . . . .	32
5.1.3	Nedre skranke for $T$ i TOP-T . . . . .	33
5.1.4	Nedre skranke for $T$ i TOP-T-k . . . . .	34
5.1.5	Minimumsverdi av $s$ i TOP-s . . . . .	34
5.1.6	Minimumsverdi av $s$ i TOP-s-k . . . . .	35
5.1.7	Minimumsverdi av $T$ i TOP-T . . . . .	35
5.1.8	Minimumsverdi av $T$ i TOP-T-k . . . . .	35
5.1.9	Nedre skranker for $L$ i TOP-L og TOP-L-k . . . . .	36
5.1.10	Skranker ved ikke-preemptiv skedulering . . . . .	37
5.2	Kompleksitetsanalyse . . . . .	37
5.2.1	$NP$ -kompletthet av TOP-L-k og TOP-L . . . . .	38
5.2.2	Polynom løsning av TOP-L-k for uvarierende trafikkmatriser . . . . .	42
5.2.3	Polynom løsning av TOP-L for $k$ -uvarierende trafikkmatriser . . . . .	43
5.2.4	$NP$ -hardhet av tilnærming bedre enn faktor $\frac{7}{6}$ for TOP-L-k . . . . .	46
5.2.5	Ikke-preemptiv skedulering . . . . .	48
<b>6</b>	<b>Algoritmer</b>	<b>49</b>
6.1	Algoritmer for TOP-s og TOP-T . . . . .	49
6.1.1	A-TOP-s: minimum antall svitsjinger . . . . .	50
6.1.2	A-TOP-T: minimum overføringstid . . . . .	55
6.2	A-TOP-L-u: minimal varighet gitt uvarierende og $k$ -uvarierende trafikkmatriser . . . . .	61
6.3	Heuristikker og tilnærmingsalgoritmer for TOP-L . . . . .	64
6.3.1	A-TOP-s og A-TOP-T som heuristikker . . . . .	64
6.3.2	A-TOP-s-mod: modifisert A-TOP-s . . . . .	64
6.3.3	H-TOP-L( $\alpha$ ) . . . . .	66
<b>7</b>	<b>Beregningsresultater</b>	<b>71</b>
7.1	Implementasjon . . . . .	71
7.2	Forekomster av uvarierende og $k$ -uvarierende trafikkmatriser . . . . .	73
7.3	Empirisk analyse av heuristikkene . . . . .	73
7.3.1	Gjennomsnittlig tilnærmingsratio, alle algoritmer . . . . .	75
7.3.2	Sammenlikning av A-TOP-s og A-TOP-s-mod . . . . .	77
7.3.3	Valg av $\alpha$ for H-TOP-L( $\alpha$ ) . . . . .	78
7.3.4	Beste heuristikk og tilnærmingsratio . . . . .	83
7.3.5	Kjøretider . . . . .	84

<b>8 Oppsummering og konklusjon</b>	<b>87</b>
8.1 Oppsummering . . . . .	87
8.2 Konklusjon . . . . .	88
8.3 Videre arbeid . . . . .	89
<b>A Programkode</b>	<b>91</b>
A.1 Bestemme (k-)uvarierende matriser . . . . .	91
A.1.1 unvarying.m . . . . .	91
A.2 Algoritme A-TOP-s . . . . .	93
A.2.1 min_s.m . . . . .	93
A.2.2 min_T.m . . . . .	94
A.2.3 dummy_v.m . . . . .	94
A.2.4 add_dummy_A_TOP_s.m . . . . .	94
A.2.5 frame_A_TOP_s.m . . . . .	95
A.2.6 s0T_A_TOP_s.m . . . . .	96
A.2.7 switching_A_TOP_s.m . . . . .	96
A.3 Algoritme A-TOP-s-mod . . . . .	97
A.3.1 mod_switchingATOPs.m . . . . .	97
A.4 Algoritme A-TOP-T . . . . .	98
A.4.1 add_dummy_ATOPT.m . . . . .	98
A.4.2 extend_ATOPT.m . . . . .	99
A.4.3 switching_ATOPT.m . . . . .	100
A.4.4 frame_ATOPT.m . . . . .	101
A.4.5 T0s_ATOPT.m . . . . .	102
A.5 Algoritme H-TOP-L( $\alpha$ ) . . . . .	103
A.5.1 frame_HTOPLalpha.m . . . . .	103
A.5.2 sT_HTOPLalpha.m . . . . .	103
A.6 Simulering . . . . .	103
A.6.1 sim_alg.m . . . . .	103
<b>Bibliografi</b>	<b>107</b>





# Kapittel 1

## Innledning

Dette første kapittelet inneholder en kort innledning til oppgaven. Vi presenterer problemområde og mål for oppgaven, og gir en oversikt over oppgavens struktur.

### 1.1 Bakgrunn og motivasjon

“The recent development of combinatorics is somewhat like a cinderella story. It used to be looked down on by “mainstream” mathematicians as being somehow less respectable than other areas [...] Then along came the prince of computer science with its many mathematical problems and needs - and it was combinatorics that best fitted the glass slipper held out.”

*Anders Björner og Richard Stanley ([3])*

“The technology improvements in algorithms, modeling languages, software and hardware have made the methodology accessible, easy to use, and fast. So the Age of Optimization has arrived.”

*George L. Nemhauser ([17])*

Studiet av telleproblemer på endelige mengder, *kombinatorikk*, er en av de eldste grenene i matematikken. Med introduksjonen av datamaskiner i forrige århundre fikk dette tidligere så teoretiske feltet et kraftig oppsving, og er nå også å regne for et viktig område innenfor den anvendte matematikken. Kombinatorikken gir i dag essensielle verktøy for konstruksjon og analyse av beregningsalgoritmer, ikke minst ved å hjelpe oss å besvare spørsmålene “Hvor vanskelig er problemet?” og “Hvor lang tid

tar det å løse det?”.

*Grafteorien* er et annet fagfelt som har gjennomgått en tilsvarende utvikling. Fra Eulers studium av broene i Königsberg i 1736 har grafstrukturer vært studert som ren og anvendt matematikk. Da det viste seg at svært mange problemer i det framvoksende informatikkfaget kunne formuleres som grafproblemer, fikk grafteorien enda større betydning som en egen gren av matematikken.

*Optimering* er studiet av minimering og maksimering av funksjoner over gitte begrensninger. Moderne optimering har sin opprinnelse i allierte militære forskningsmiljøer under og etter andre verdenskrig. Man utviklet her et metodeapparat for analyse av militære og sivile operasjoner, i den hensikt å maksimere nytte og minimere kostnader i en rekke forskjellige problemer. Navnet *operasjonsanalyse* (eng. “*operations research*”) henger ennå igjen som navnet på deler av moderne optimering. Det er i dag en voldsom utbredelse av optimering i alle vitenskaper og virksomheter, og matematikeren George L. Nemhauser kan ha rett i at vi i dag lever i optimeringens tidsalder, “the Age of Optimization”([17]).

Alle disse tre fagområdene, som modnet til i løpet av det 20. århundret, finner sammen i den kanskje mest tidstypiske anvendelsen ved overgangen til det 21. århundret: digital tele - og datakommunikasjon. Kommunikasjonsnettverkene er ingenting om ikke inkarnasjonen av grafer, og for å kunne utnytte kapasiteten på best mulig måte har vi behov for nettopp *kombinatorisk optimering*.

I denne oppgaven skal vi undersøke hvordan vi raskest mulig kan overføre digital trafikk i en spesiell type satellittsystemer. For å analysere og løse dette problemet vil vi anvende begreper og metoder fra kombinatorikk, grafteori og optimering.

## 1.2 Problem

De fleste kommunikasjonssystemer har den fellesnevneren at de har begrensede ressurser tilgjengelig, f.eks. i form av liten båndbredde eller få antenner. For å la et stort antall brukere kommunisere simultant over et system med begrensede ressurser finnes det flere forskjellige metoder. En av disse er *Time Division Multiple Access*, TDMA. Denne oppgaven tar for seg et optimeringsproblem som oppstår i satellittsystemer som benytter TDMA-metoden, *Satellite-Switched Time Division Multiple Access* (SS/TDMA) satellittsystemer. Metoden består i korthet av at man deler opp tiden i små tidsluker, og i hver luke får et par av sender og motta-

ker kommunisere over hver antenne i satellitten. I neste luke overlates antennene til nye par av sendere - og mottakere, og slik går tilgangen til de begrensede ressursene på tur.

Gitt et mengde trafikk som skal overføres i systemet, ønsker vi å finne et sendeskjema som formidler all trafikken i løpet av kortest mulig tid. Denne tiden er en sum av alle tidslukene og tidskostnadene som påløper når antennene endrer retning mellom par av sendere og mottakere. Oppgavens hovedtema er optimeringsproblemet som består i å finne et slikt skjema med minimal varighet, *SS/TDMA trafikkoptimeringsproblemet*.

### 1.3 Oppgavens mål

Et av oppgavens mål er å gi et overblikk over de viktigste resultatene fra litteraturen som omhandler SS/TDMA trafikkoptimeringsproblemet. De ulike resultatene vil her bli presentert i en felles notasjon, og bevisene vil gjengis på et mer utfyllende og noe lavere nivå enn i litteraturen. Det tilstrebes således å gi en enhetlig presentasjon av problemområdet, slik at oppgaven kan tjene som en introduksjon til SS/TDMA trafikkoptimering for den interesserte leser med noe bakgrunn i matematikk og informatikk. Grunnleggende teori blir også presentert, for å gjøre oppgaven mest mulig tilgjengelig og selvstendig.

I store deler av litteraturen har man konsentrert seg om et spesialtilfelle av optimeringsproblemet, der antall antenner er lik det minimale antallet sendere eller mottakere. Både algoritmer og kompleksitetsanalyse har utgangspunkt i dette spesialtilfellet. Det er i denne oppgaven en målsetning å generalisere både analyse og algoritmer til et mer generelt tilfelle, der antallet antenner kan være mindre eller lik det minimale antallet sendere eller mottakere.

Heuristikkene som presenteres i oppgaven har ingen bevisbart god tilnærming til en optimal løsning av trafikkoptimeringsproblemet. Vi foretar i oppgaven en empirisk analyse av heuristikkenes gjennomsnittlige tilnærmingsratio, og ønsker på bakgrunn av analysen å kunne utpeke den beste heuristikken.

### 1.4 Oppgavens struktur

#### Teori

I kapittel 2 presenteres det teoretiske fundamentet for oppgaven. Vi skal gi de grunnleggende definisjonene i optimering og grafteori, samt flere

kjente resultater som benyttes i oppgaven. Vi skal også forklare kompleksitetsanalyse av algoritmer og problemer.

### **Problem**

Kapittel 3 gir en nøyere forklaring av SS/TDMA trafikkoptimeringsproblemet.

### **Matematisk modell**

Den matematiske modellen vi anvender for å representere problemet gis i kapittel 4.

### **Analyse**

I kapittel 5 gir vi en teoretisk analyse av problemet. Det presenteres en rekke kjente resultater om nedre skranker og minimumsverdier for problemet, samt en kompleksitetsanalyse av problemet.

### **Algoritmer**

Analysen fra kapittel 5 gir opphav til flere algoritmer og heuristikker som presenteres i kapittel 6.

### **Beregningsresultater**

I kapittel 7 ser vi på kvaliteten av heuristikkene fra kapittel 6, ved å foreta en empirisk analyse av hvor godt heuristikkene tilnærmer en optimal løsning av problemet.

### **Oppsummering og konklusjon**

Analysen og beregningsresultatene oppsummeres i kapittel 8. Det gis en konklusjon på bakgrunn av funnene gjort i oppgaven, og vi skisserer så videre arbeid i feltet.

### **Programkode**

I tillegg A ligger programkoden for implementasjonen av algoritmene vedlagt.

## Kapittel 2

# Teori

I dette kapittelet presenterer vi de viktigste teoretiske byggesteinene for oppgaven. Teorien omfatter fundamentale begreper og resultater fra optimering, grafteori og kompleksitetsanalyse.

### 2.1 Optimering

Optimering er generelt sett studiet av minimering og maksimering av funksjoner. Definisjonen av et *optimeringsproblem* er som følger:

**Definisjon 1.** Et optimeringsproblem  $P$  er et problem

$$\text{maksimer}\{f(x) : x \in A\}$$

der  $f : A \rightarrow \mathbb{R}$  er en gitt funksjon definert på en mengde  $A$  ([8]).

Vi kan også på tilsvarende måte definere et minimeringsproblem. Hvert punkt i  $A$  kalles en *tillatt løsning*. Funksjonen  $f$  kalles problemets *objektivfunksjon*.

Et punkt  $x^*$  er en *optimal løsning* av problemet  $P$  dersom  $f(x^*) \geq f(x)$  for alle  $x \in A$ . Den *optimale verdien* av  $P$  defineres som *supremum* av verdien av  $f(x)$  for alle  $x$  i  $A$ ,  $\sup\{f(x) : x \in A\}$ .

En mindre klasse optimeringsproblemer er de *kombinatoriske optimeringsproblemene*. La  $E$  være en endelig mengde, og  $\mathbf{F}$  en familie av delmengder av  $E$ .  $w : E \rightarrow \mathbb{R}_+$  er en ikke-negativ funksjon, og  $w(F) = \sum_{e \in F} w_e$  for hver  $F \in \mathbf{F}$ .

**Definisjon 2.** Et kombinatorisk optimeringsproblem er et problem

$$\text{maksimer}\{w(F) : F \in \mathbf{F}\}$$

For mer optimeringsteori henviser vi til “Combinatorial optimization” av W. Cook et al. [6].

## 2.2 Grafteori

Mange optimeringsproblemer innebærer å lete etter bestemte strukturer i *grafer*. Vi skal her gi korte definisjoner av de viktigste begrepene i grafteori, hentet fra [8].

En *graf* er et ordnet par  $G = (V, E)$ , der  $V$  er en endelig mengde med elementer og  $E$  en endelig mengde av uordnede par av elementer fra  $E$ .  $V$  kalles *nodene* i grafen, og  $E$  kalles grafens *kanter*. I en *vektet graf* finnes det en vektfunksjon  $w(e) : E \rightarrow \mathbb{R}_+$  som gir hver kant i grafen en ikke-negativ verdi. En kant  $e$  i  $E$  skriver vi ofte som  $e = uv$ , der  $uv$  er det ordnede paret av noder i  $V$ . Vi sier da at  $u$  og  $v$  er *inntil* kanten  $e$ , og omvendt. Antall kanter inntil en node utgjør nodens *grad*. En *vei* i en graf er en sekvens  $v_0, e_0, v_1, e_1, \dots, v_{n-1}, e_{n-1}, v_n, e_n, v_{n+1}$  av noder  $v_i$  og kanter  $e_i = v_i v_{i+1}$  for  $i = 1, \dots, n$ .

I en *bipartitt* graf kan vi dele nodene i to delmengder  $V_1$  og  $V_2$ , slik at alle kanter er inntil én node i hver av mengdene. Vi vil i oppgaven anvende bipartitte grafer i den matematiske modellen av satellittsystemet.

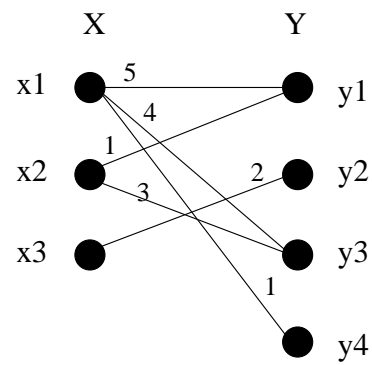
Vi assosierer ofte grafer med visualiseringen av grafstrukturen i  $\mathbb{R}^2$ , som vi kaller grafens *embedding*. I en embedding av en graf tegner vi nodene som punkter og hver kant som en kurve som forbinder to noder. Figur 2.1 viser en embedding av en vektet bipartitt graf med 7 noder og 6 kanter.

### 2.2.1 Matchinger

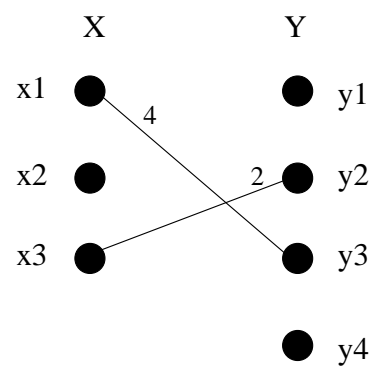
*Maksimum matchinger* i bipartitte grafer er et sentralt tema i denne oppgaven. En *matching* i en graf  $G = (V, E)$  er en mengde  $M$  av kanter slik at ingen node i  $G$  er inntil flere enn én kant i  $M$ . Vi sier at en matching *dekker* en node dersom en kant i matchingen er inntil noden. Figur 2.2 viser en matching i den vektete bipartitte grafen fra figur 2.1

En *maksimum* matching i en graf er en matching av maksimum kardinalitet, det vil si at den dekker flest mulig noder. En *komplett matching* er en matching som dekker alle nodene.

Vi kan for en matching  $M$  i en graf  $G$  definere en *M-tilføyende* vei i graf-



Figur 2.1: Vektet bipartitt graf.



Figur 2.2: Matching i vektet bipartitt graf.

en. En  $M$ -tilføyende vei er en vei der enhver kant tilhører matchingen  $M$ , og hvor endenodene i veien ikke er dekket av  $M$ . Eksistensen av en slik vei kan brukes til å vise at det finnes en maximum matching, slik følgende teorem fra [6] viser. For bevis henviser vi til [6].

**Teorem 1.** *En matching  $M$  i en graf  $G$  er en maksimum matching hvis og bare hvis det eksisterer en  $M$ -tilføyende vei i grafen.*

Dette teoremet antyder en mulig fremgangsmåte for å konstruere en maksimum matching. Ved å teste alle mulige matchinger  $M$  i grafen for eksistens av en  $M$ -tilføyende vei, vil vi til slutt finne en maksimum matching. I verste tilfelle kan vi risikere å måtte teste *alle* mulige matchinger i grafen, og dette vil derfor tjene svært dårlig som en algoritme i praksis. Det finnes en rekke maksimum matching algoritmer som bygger på prinsippet i Teorem 1, men som vil gi overkommelige kjøretider. En av de mest kjente av disse er *den ungarske metoden* ([15]), som finner maksimum matchinger i *bipartitte* grafer. Hovedideen bak denne algoritmen er å starte med en vilkårlig matching  $M$ , og fra den  $M$ -tilføyende veien finne en ny og større matching. Dette gjentas inntil det ikke finnes noen  $M$ -tilføyende vei. Vi har da en maksimum matching.

Det blir i denne oppgaven ikke gitt en nærmere forklaring av den ungarske metoden eller andre algoritmer for å finne maksimum matching. Vi vil i stedet fokusere på trafikkoptimeringsalgoritmer som *anvender* maksimum matching algoritmer, for å minimere varigheten av TDMA skjemaer. For en omfattende behandling av matchingteori og -algoritmer, vil vi henvise til [15], [14] og [16].

I algoritmene A-TOP-s og A-TOP-T i kapittel 6 anvender vi maksimum matching algoritmer for å finne *komplette* matchinger. Dersom det finnes en komplett matching i en graf, vil denne blir funnet av en maksimum matching algoritme. Men når kan vi garantere at det finnes en komplett matching i en graf? Vi skal presentere et teorem og et korollar av dette teoremet som gir oss en mulighet til å bevise at det finnes en komplett matching. Bevisene finnes i [15]. Hvis  $X$  er en mengde i nodemengden  $V(G)$  for en graf  $G$ , la  $\Gamma(X)$  betegne alle nodene i  $V(G)$  som er inntil minst en node i  $X$ .

**Teorem 2.** *La  $G = (A, B)$  være en bipartitt graf. Da har  $G$  en matching av  $A$  inn i  $B$  hvis og bare hvis  $|\Gamma(X)| \geq |X|$  for alle  $X \subseteq A$ .*

Dette teoremet har en rekke forskjellige ekvivalente former formulert av forskjellige opphavspersoner, og denne utgaven av teoremet er av P. Hall



([15]). Et korollar formulert av Frobenius kalles *giftemålsteoremet* (*the Marriage Theorem*), og karakteriserer bipartitte grafer med en komplett matching ([15]).

**Korollar 1.** *En bipartitt graf  $G = (A, B)$  har en komplett matching hvis og bare hvis  $|A| = |B|$  og  $|\Gamma(X)| \geq |X|$  for alle  $X \subseteq A$ .*

### 2.2.2 Litt matriseteori

Algoritmen A-TOP-T i kapittel 6 forklares fra matriserepresentasjonen av trafikkoptimeringsproblemet, og vi foklærer derfor kort et resultat for matriser som er ekvivalent med korollar 1 over.

Vi skal først definere noen begreper. En *dobbeltstokastisk* matrise er en matrise der alle rad - og kolonnesummer er lik 1. En *kvasi-dobbeltstokastisk* matrise er en kvadratisk matrise der alle rad - og kolonnesummene er like. En *system av distinkte representanter* i en matrise er en mengde positive elementer i marisen slik at ingen av elementene ligger på samme rad eller kolonne.

Hvis  $X$  er mengden av linjer i en matrise  $D$ , la  $\Gamma(X)$  betegne alle linjene som har et positivt element i minst en av linjene i  $X$ . Følgende korollar er ekvivalent med korollar 1:

**Korollar 2.** *En matrise  $D$  har et distinkt system av representanter som dekker alle rader og kolonner i  $D$  hvis og bare hvis  $D$  er en kvadratisk matrise og  $|\Gamma(X)| \geq |X|$  for alle  $X \subseteq A$ .*

Av dette følger igjen neste korollar([4]):

**Korollar 3.** *En  $N \times N$  kvasi-dobbeltstokastisk matrise har et system av distinkte representanter bestående av  $N$  elementer.*

## 2.3 Problem - og algoritmeanalyse

Når vi ønsker å løse et problem er vi naturlig nok interessert i å vite om problemet er lett eller vanskelig å løse, og om det i det hele tatt lar seg løse. I konteksten av problemer vi ønsker å løse ved hjelp av data-maskiner ser vi her bort fra klassen av problemer som anses som bevist uløselige. Begrepene *lett* og *vanskelig* er da gjerne knyttet til hvor lang tid det tar å løse problemet.

Hvis vi ønsker å bestemme hvor lang tid en algoritme bruker på løse

et problem kan vi gjøre dette på to måter. Vi kan gjøre en *empirisk analyse* av algoritmen eller en *verste-tilfelle kompleksitetsanalyse*.

En empirisk analyse vil gi et estimat på hvor lang tid algoritmen bruker på å løse problemet i praksis, bygget på hvor lang tid den bruker på en eller flere representative instanser av problemet. En slik analyse vil være avhengig av hasigheten på datamaskinen den kjøres på, og den sier ikke nødvendigvis noe om hvor lang tid algoritmen i verste tilfelle kan bruke på å løse problemet.

Kompleksitetsanalysen av en algoritme gir oss et uttrykk for hvordan tidsforbruken kan øke som funksjon av størrelsen på problemet. I motsetning til den empiriske analysen, er dette en verste-tilfelle analyse som både angir hvor lang tid problemet maksimalt kan bruke, og hvorvidt en algoritme er raskere enn en annen. Gjennom kompleksitetsanalysen kan vi dele opp algoritmene i forskjellige klasser ut fra verste-tilfelle hastighet.

For et gitt problem er vi interessert i hvor raske algoritmer det finnes for å løse problemet. Klassene av algoritmehastigheter gir opphav til et formalisert konsept om inndeling i lette og vanskelig løsbare problemer, basert hvor raske algoritmer det finnes for å løse problemet. Kompleksitetsanalyse av et problem består i å klassifisere problemet etter denne inndelingen. Vi skal under forklare nærmere kompleksitetsanalyse av både algoritmer og problemer.

### 2.3.1 Kompleksitet

Kompleksitetsanalysen av en algoritme gir en verste-tilfelle analyse av hvor lang tid algoritmen bruker. Dette gjøres ved å angi en funksjon som gir en øvre skranke for hvor raskt antall beregningssteg som gjøres av algoritmen øker som funksjon av størrelsen på problemet. Funksjonen angis med såkalt O-notasjon, som vi skal definere her. Fra [20] har vi:

**Definisjon 3.**  $T(N) = O(f(N))$  dersom det finnes positive konstanter  $c$  og  $n_0$  slik at  $T(N) \leq c f(N)$  når  $N \geq n_0$ .

Vi angir så kompleksiteten til en algoritme ved å angi med notasjonen over verste-tilfellet antallet beregningssteg som algoritmen gjør for å løse problemet. Vi kan f.eks. si at en algoritme er  $O(N^2)$ , der  $N$  er en verdi som angir størrelsen av mengden input-data til problemet. Det betyr da at det finnes en kvadratisk funksjon av  $N$ , som begrenser oppad hvor raskt tidsforbruket til algoritmen øker når  $N$  øker. Man oppgir i O-notasjon ikke konstant- eller laveregradsledd i funksjonen  $f(N)$ , slik

at  $O(2N^2 + N)$  oppgis som  $O(N^2)$ .

Algoritmer av kompleksitet  $O(f(N))$ , der  $f(N)$  er polynom funksjon av problemets størrelse, kalles en *polynom algoritme*. Disse lar seg løse av dagens datamaskiner med tidsforbruk som gjør dem anvendbare. Det finnes algoritmer der økningen av tidsforbruket ikke lar seg begrense av en polynom funksjon, slike kaller vi *eksponensiell tids-algoritmer*. Et eksempel på en slik kompleksitet er  $O(2^n)$ . Ved store problemer kan vi risikere tidsforbruk for slike algoritmer som medfører at det aldri kan produseres nok regnekraft i verden til å løse dem.

Det finnes også en klasse algoritmer som ligger mellom de to overnevnte i tidsforbruk. For de *pseudopolynome* algoritmene er  $f$  ikke bare en funksjon av størrelsen  $N$  av input-data, men også *verdiene* av input-data ([16]). Dette kan gi et uforutsigbart økende tidsforbruk for instanser av problemet med høye input-verdier.

### NP og NP-komplettethet

Når vi deler inn problemer i lette og vanskelig løsbare problemer er det nettopp eksistensen av en polynom algoritme som benyttes som kriterium. Klassen  $P$ , de *polynomt løsbare problemene*, består av alle problemer som vi vet lar seg løse av polynome algoritmer. Disse utgjør de lette problemene.

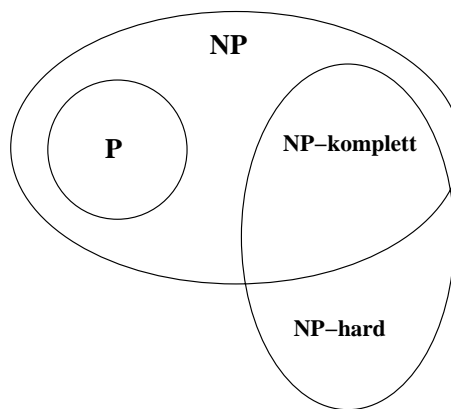
Det finnes også en større klasse problemer som vi kaller *ikke-deterministiske polynome problemer*,  $NP$  (*nondeterministic polynomial time*). Disse problemene vil la seg løse i polynom tid av en imaginær ikke-deterministisk maskin, som alltid gjør et optimalt neste valg dersom dette valget fører mot en løsning av problemet. Klassen  $NP$  inkluderer også alle polynomt løsbare problemer.  $P$  er en delmengde av  $NP$ .

I  $NP$  finnes også en annen delmengde av problemer, de  $NP$ -komplette problemene. Disse er per definisjon de vanskeligste problemene i  $NP$ . Dette fordi et  $NP$ -komplett problem defineres av egenskapen at ethvert annet problem i  $NP$  kan bli *polynomt redusert* til det.

Et problem  $P_1$  kan bli polynomt redusert til et annet problem  $P_2$ , på følgende måte: Det finnes en transformasjon som avbilder enhver instans av  $P_1$  til en instans av  $P_2$ , slik at en løsning av  $P_2$  kan avbildes tilbake til en løsning av  $P_1$ . Transformasjonen må være en polynom algoritme. Notasjonen for at  $P_1$  kan polynomt reduseres til  $P_2$  er slik:

$$P_1 \propto P_2$$

En annen klasse er de *NP*-harde problemene. Disse problemene er ikke nødvendigvis i *NP*, men det gjelder som for de *NP*-komplette problemene at ethvert problem i *NP* kan bli polynomt redusert til et *NP*-hardt problem. *NP*-komplette problemer er de *NP*-harde problemene som også er i *NP*. *NP*-harde problemer er dermed minst like vanskelige som *NP*-komplette.



Figur 2.3: NP

Definisjonen av de *NP*-komplette problemene impliserer at dersom et *NP*-komplett problem kan løses i polynom tid, kan også alle de andre problemene i *NP* løses i polynom tid. Da vil vi ha at  $P = NP$ . Det antas fortsatt at  $P \neq NP$ , da ingen har klart å vise det motsatte ved å finne en polynom algoritme for et *NP*-komplett problem.

Teorien om *NP*-kompletthet gir oss muligheten til å bevise at et problem ikke er i *P*, antatt at  $P \neq NP$ . Dette er svært nyttig i analysen av problemer. Dersom vi klarer å vise at et problem er *NP*-komplett, behøver vi ikke lete etter polynome algoritmer for problemet. Slike vil ikke eksistere med mindre  $P = NP$ . Siden vi i vanlige tilfelle ikke vil ønske å bruke eksponensiell-tids algoritmer, vil vi betrakte den optimale løsningen av *NP*-komplette problemer som uoppnåelig.

For å vise at et problem  $P_2$  i *NP* er *NP*-komplett, må vise at at et kjent *NP*-komplett problem  $P_1$  kan reduseres til det, som angitt over. Et eksempel på en vanlig reduksjon er å vise at  $P_2$  inneholder et *NP*-komplett problem  $P_1$  som et spesialtilfelle.

Selv om vi har bevist at et problem generelt sett er *NP*-komplett, kan

det fortsatt hende at det eksisterer klasser av input-data for hvilke det eksisterer polynome algoritmer. For at de skal ha innvirkning på hvordan vi løser problemet i praksis må de dog være lette å identifisere, samt omfatte en viss prosentandel av de reelle input-data.

### 2.3.2 Heuristikker og tilnærmingsalgoritmer

I mangel på optimale løsninger i polynom tid, kan vi for *NP*-komplette problemer designe algoritmer som vi håper finner *gode* løsninger i polynom tid. En slik algoritme kaller vi en *heuristikk* ([6]).

En heuristikk bygger ofte på en intuisjon om hvordan vi finner gode resultater. For noen heuristikker kan vi garantere at den gode løsningen avviker mindre enn viss faktor fra en optimal løsning. Disse kaller vi *tilnærmingsalgoritmer*.

#### Positive kompleksitetsresultater

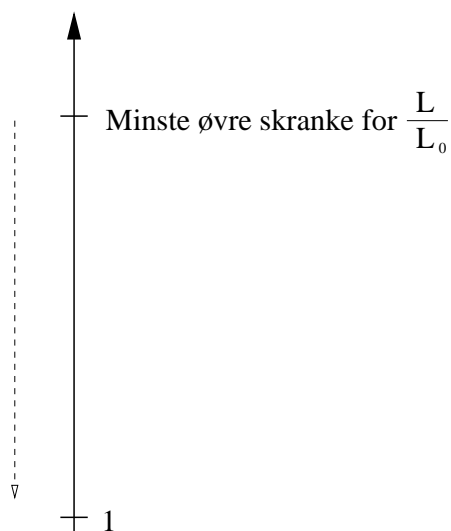
Vi definerer for en heuristikk *tilnærmingsratioen* som løsningen  $L$  av problemet vi får fra heuristikken dividert med verdien  $L_0$  av en optimal løsning:

$$\frac{L}{L_0}$$

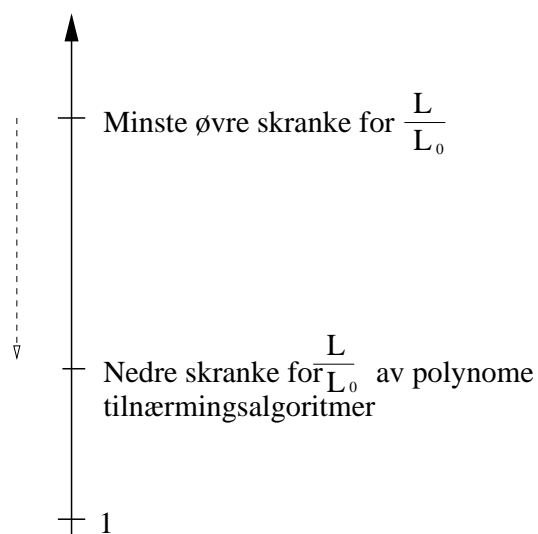
Jo nærmere 1 en slik tilnærmingsratio er, dess bedre tilnærming til en optimal løsning gir heuristikken. Se figur 2.4 for en illustrasjon.

For tilnærmingsalgoritmene kan vi altså garantere at tilnærmingsratioen er mindre enn en øvre skranke. Eksistensen av en slik *bevisbart* god tilnærming av et problem er det vi kaller et *positivt* kompleksitetsresultat for tilnærming. I analysen av *NP*-komplette problemer vil vi lete etter tilnærmingsalgoritmer med best mulig øvre skranke på tilnærmingsratioen.

For heuristikker vil vi foreta en empirisk analyse, og måle hvor lave tilnærmingsratioer vi får i de gjennomsnittlige tilfellene. Dette vil gi et probabilistisk estimat av ratioen. En slik empirisk analyse kan vi også foreta for tilnærmingsalgoritmene, da den gjennomsnittlige tilnærmingen ofte kan være langt bedre enn verste-tilfellet garantien.



Figur 2.4: Øvre skranke for tilnærmingsratio.



Figur 2.5: Øvre og nedre skranke for tilnærmingsratio

### Negative kompleksitetsresultater

I jakten på bevisbart gode tilnærmingsalgoritmer oppstår naturlig nok spørsmålet om *hvor* gode tilnærminger vi kan finne i polynomisk tid. Vi kan for svært mange problemer vise at det fins en nedre skranke for hvor gode tilnærmingsratioer vi kan garantere for problemet. Dette er et

*negativt* kompleksitetsresultat for tilnærming. Se figur 2.5.

En vanlig teknikk for å bevise et det finnes en slik skranke er den såkalte *gap technique*. Denne teknikken bygger på en reduksjon fra en *ja-nei formulering* av et *NP*-hardt problem. En ja-nei formulering av et minimeringsproblem kan lages på følgende måte: Hvis det opprinnelige problemet lyder "Finn en løsning som minimerer  $L$ ", så vil ja-nei formuleringen av det samme problemet bli "Finnes det en løsning  $\leq d$ ?" for en verdi  $d$ .

Anta at vi har et *NP*-hardt problem  $P_2$ . Vi ønsker da å finne en polynom reduksjon av ja-nei formuleringen  $P_2$  til minimeringsproblemet  $P_1$ , slik at det følgende gjelder:

- Instanser av  $P_2$  med svar *ja* reduseres til instanser av  $P_1$  med verdi av objektivfunksjonen  $\leq c^*$ .
- Instanser av  $P_2$  med svar *nei* reduseres til instanser av  $P_2$  med verdi av objektivfunksjonen  $\leq g \cdot c^*$ , for et fast reellt tall  $g \geq 1$ .

Det følger da at det ikke eksisterer en tilnærmingsalgoritme for  $P_1$  med garantert tilnærmingsratio  $< g$ , med mindre  $P = NP$  ([11]).

For flere resultater om kompleksitet og tilnærming henvises det til [2].





## Kapittel 3

# Problem

Vi gir i dette kapittelet en innføring i SS/TDMA trafikkoptimeringsproblemet, og diskuterer bakgrunn og problemformulering.

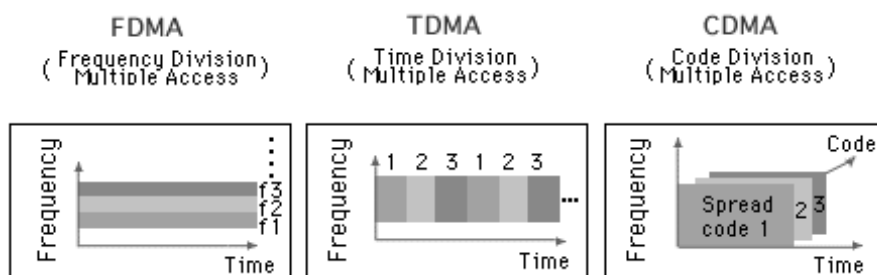
### 3.1 Bakgrunn

Et sentralt problem i digital telekommunikasjon er fordeling av tilgjengelig båndbredde til simultane brukere av kommunikasjonssystemer. Det må ofte benyttes *multiplexing*, dvs. at to eller flere brukere samtidig sender signaler over en *transponder*. En transponder kan være et rent fysisk medium, som en signalkabel eller en antenne som mottar, forsterker og sender signalet videre, eller en tilgjengelig del av radiofrekvensspekteret. Satellitt - og mobiltelefonisystemer er eksempler der mange brukere med geografisk spredning ønsker å kommunisere med hverandre. Kommunikasjonen skjer gjennom transpondere som av tekniske og økonomiske årsaker er færre enn antall ønskede simultane signaloverføringer. Man må derfor finne måter å effektivt la en stor mengde distinkte sendere og mottakere dele et lite antall transpondere.

Det benyttes i hovedsak tre multiplexing metoder: *Frequency Division Multiple Access* (FDMA), *Code Division Multiple Access* (CDMA) og *Time Division Multiple Access* (TDMA).

#### 3.1.1 FDMA

FDMA er den eldste teknologien, og selv om den kan brukes til multiplexing av digitale signaler har den vært mest utbredt i analoge mobiltelefonisystemer. I FDMA deles frekvensbåndet opp i intervaller og signalene til forskjellige brukere separeres ved å sende signalene samtidig i forskjellige frekvensintervaller.



Figur 3.1: Multiplexing metoder

### 3.1.2 CDMA

En nyere multiplexing metode som vinner terreng både i mobiltelefoni og satellittkommunikasjon er CDMA. Denne metoden er basert på at alle signalene sendes samtidig over hele det tilgjengelige frekvensbåndet. Signalene moduleres av sender med en digital signatur, en bruker-spesifikk bølgeform, og forkastes eller godtas av mottaker på bakgrunn av denne. En variant av CDMA vil bli brukt av den tredje generasjons mobile nettverksteknologien UMTS.

### 3.1.3 TDMA

I TDMA deles tiden i intervaller og de ulike signalene separeres ved å plasseres i hver sine sekvensielle tidsintervall. I kort tid dedikeres dermed hele frekvensbåndet til ett signal, før neste signal overtar. For en illustrasjon av TDMA og de andre multiplexing metodene, se figur 3.1.

For tiden er TDMA den mest etablerte og utbredte av multiplexing metodene. Utbredelsen skyldes i stor grad at metoden benyttes store internasjonale standarder for mobiltelefoni. Den europeiske standarden GSM, den amerikanske NADC og den japanske JDC benytter alle varianter av TDMA. Slike varianter vil f.eks. innebære kombinasjon av FDMA og TDMA, ved at frekvensbåndet deles i kanaler og det benyttes TDMA i hver av disse kanalene.

## 3.2 SS/TDMA satellittsystemer

TDMA, alene og i kombinasjon med FDMA og CDMA, benyttes i svært mange forskjellige satellittsystemer i dag, fra satellittelefoni til systemer for lavhastighetsformidling av kredittkortverifikasjon. Satellittkommunikasjon stiller store krav til fart og kapasitet på overføringene. Bru-

kerne er mange, de har ofte stor geografisk spredning og de forventer rask og kontinuerlig overføring av sin digitale trafikk. Kommunikasjonen skjer via antenner på satellitten som mottar og videresender trafikk. I enda større grad enn i bakkebaserte mobiltelefonisystemer er antallet slike transpondere en flaskehals. Hensyn som bl.a. kostnadene per vekt- og volumenhet ved satellittoppskytninger fører til at det er svært få antenner, og en god multiplexing strategi er derfor svært viktig.

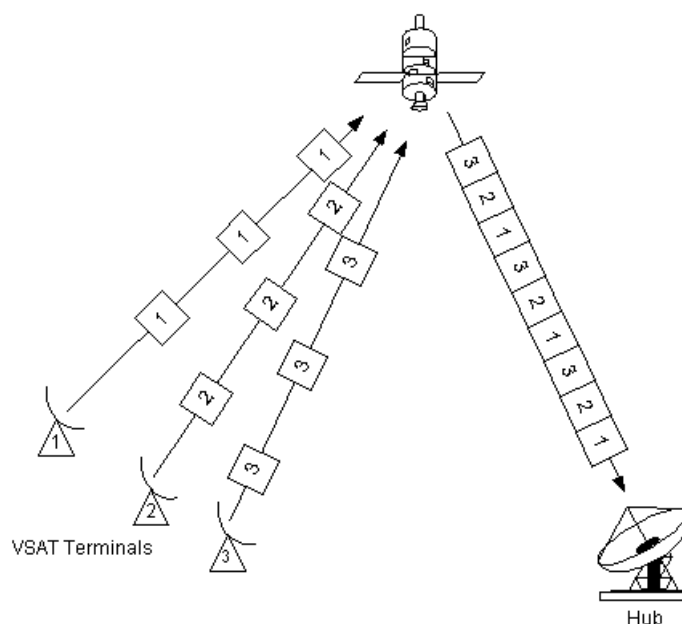
Jeg ønsker i denne oppgaven å fokusere på et optimeringsproblem som oppstår i et forenklet satellittsystem som benytter TDMA, et *Satellite-Switched Time Division Multiple Access* (SS/TDMA) system. Systemet består totalt av et antall sendestasjoner, et antall mottakerstasjoner og en satellitt med retningsorienterte, motorstyrte antenner som i en gitt posisjon kun kan formidle signaler fra én sender til én mottaker. Det er færre antenner enn både sendere og mottakere, og dette antallet antenner bestemmer det maksimale antall simultane koblinger mellom sendere og mottakere. Multiplexing-problemet som oppstår når alle sendere og mottakere skal dele de tilgjengelige antennene, løses av en datamaskin ombord i satellitten ved hjelp av TDMA metoden.

Tiden deles i intervaller, og i et tidsintervall er hver antenne dedikert til ett sender/mottaker-par. For å bytte sender eller mottaker (eller begge) for en antenne, må antennen skifte retning. Dette kalles en *svitsjing*, og tar en bestemt tid. Antennenes faste posisjon og retning mellom svitsjingene kalles en *svitsjtilstand*. Alle svitsjinger må skje samtidig, men alle antennene må ikke nødvendigvis foreta en endring av svitsjtilstanden hver gang.

### 3.3 SS/TDMA trafikkoptimeringsproblemet

Gitt et SS/TDMA satellittsystem som beskrevet over, tar vi utgangspunkt i at det eksisterer en mengde trafikk som skal overføres gjennom systemet, systemets *trafikkbehov*. Trafikken er beskrevet ved antall tidsenheter digitale signaler som skal overføres mellom hvilke par av sende- og mottakerstasjoner. Målet er en effektiv plan, et *TDMA skjema*, for formidling av den samlede trafikken fra alle sendere til alle mottakere. Skjemaet vil beskrive en sekvens av svitsjtilstander og svitsjtilstandenes varighet. Et *gyldig* skjema er et skjema som gir en overføring av hele trafikkmengden, uten å ta stilling til tiden eller antall svitsjinger som kreves før alle signaler er overført til de korrekte mottakerstasjonene. Deler av litteraturen kaller et slikt skjema vekselvis også en *TDMA frame*.

Varigheten av et skjema, gitt et trafikkbehov, bestemmes av både an-



Figur 3.2: Eksempel på SS/TDMA system: VSAT (Very Small Aperture Terminal) satellittsystem for banktransaksjoner. Systemet har 3 sendere, 1 mottaker og 1 antenne.

tallet svitsjinger og varigheten av svitsjtilstandene. Ved å summere varighetene av alle svitsjtilstandene og den samlede tiden det tar å endre antennekonfigurasjonen under alle svitsjingene, fås den totale varigheten av TDMA skjemaet.

For hver gitte trafikkmengde vil det kunne eksistere flere gyldige TDMA skjemaer, med svitsjtilstander av ulik tidslengde og ulikt antall svitsjinger. Problemet vi ønsker å fokusere på i denne oppgaven er å finne skjemaet som overfører all trafikk på kortest mulig tid. I konstruksjonen av et slikt skjema vil vi forsøke å finne en sekvens av svitsjtilstander som minimerer tiden systemet faktisk bruker på å overføre signalene. Men for å minimere den totale varigheten må vi i tillegg til svitsjtilstandenes lengde også ta hensyn til antall svitsjinger, og dermed den samlede tiden det tar å endre antennes retning.

Vi kan formulere problemet slik:

*Gitt et trafikkbehov i et SS/TDMA satellittsystem, finn en TDMA frame med minimal varighet.*

Problemet kan også formuleres ekvivalent på andre måter, f.eks. som "Gitt et trafikkbehov i et SS/TDMA satellittsystem, minimer varigheten over mengden av gyldige TDMA skjemaer". Hovedpoenget som gjenstår er dog dette: Problemet vårt er et optimeringsproblem som består av å finne den minimale totale varigheten av en TDMA frame gitt at visst trafikkbehov, og det konkrete skjemaet som oppnår denne minimale varigheten.

I denne oppgaven velger vi å kalle dette problemet *SS/TDMA trafikkoptimeringsproblemet*, eller bare *trafikkoptimeringsproblemet*. I litteraturen har dette og tilsvarende problemer blitt gitt flere forskjellige navn. Disse reflekterer de generelle klassene av problemer forfatteren(ene) anser problemet å tilhøre, eller gjenspeiler teknikker brukt i algoritmer som benyttes for å løse problemet.

En viktig klasse av problemer er *skeduleringsproblemene* (eng. scheduling problems). Disse involverer alle en planlegging av deling av ressurser, være det seg mannskap på en flyplass til flere avganger eller CPU-tid i en datamaskin til flere samtidige prosesser. Crescnzi, Xiaotie og Papdimitriou kaller i [7] problemet *the preemptive bipartite scheduling problem* (navnet har også blitt benyttet tidligere i andre artikler). Trafikkoptimeringsproblemet er et skeduleringsproblem der ressursene som skal deles er de tilgjengelige antennene. Resten av navnet sier noe om egen-skaper ved problemet som vil bli nærmere belyst i modell-kapittelet.

Et mye benyttet navn på problemet er *the timeslot assignment problem*, "tidsluke tilordningsproblemet", som hos Bongiovanni, Coppersmith og Wong i [4]. Tilordningsproblemene tar utgangspunkt i mulige tilordninger mellom par av elementer i to disjunkte mengder, der hver tilordning har en kostnad. Man ønsker å gi alle elementene en tilordning, men til lavest (eller høyest) mulig kostnad. Et kjent eksempel er "gitemålsproblemet", der de mulige tilordningene er koblinger mellom kvinner og menn med potensiale som ektefeller. Kostnaden på hver tilordning er en størrelse som forteller hvor lykkelige de blir. Gitemålsproblemet er så å sørge for at alle blir gift, og at den totale lykken for gruppen bli høyest mulig. I en TDMA frame vil hver svitsjtilstand være en tilordning av ledige tidsluker til par av sender og mottakerstasjoner, og kostnaden er varigheten av tilstanden. For å finne det optimale skjemaet vil vi søke å finne en sekvens av tilordninger med minimal total kostnad.

### 3.4 Preemption

I de fleste behandlinger av SS/TDMA satelittsystemer antas det at hvert par av sender og mottaker godtar at ikke all den ønskede trafikken overføres når de først har fått tilgang til en antenne, men at de derimot kan kastes ut for å gi adgangen til et annet par. Dette kalles *preemptiv* skedulering. Begrepet *preemption* betyr å ta kontroll over en ressurs før noe(n) andre er ferdig med den, og henspiller på at det etterfølgende paret av stasjoner får ta over tilgang til antennen før det forrige paret var ferdig med sin overføring. I operativsystemteori er det f.eks. mye fokus på forskjellene mellom preemptiv og ikke-preemptiv CPU-skedulering.

En føring man kan legge på SS/TDMA trafikkoptimeringsproblemet er at å ikke tillate preemption. Med ikke-preemptiv skedulering får vi et annet problem med noen andre egenskaper som kan være verdt å se på. Ved å legge denne føringen også på de to delprobemene får vi tilsammen 6 problemer. Flere av disse har vidt forskjellige egenskaper, som vi kommer tilbake til i analysen i Kapittel 5.

Der det ikke nevnes spesielt at vi ser på det ikke-preemptive tilfellet antas det i denne oppgaven at det benyttes preemptiv skedulering.

## Kapittel 4

# Matematisk modell

Den matematiske modellen vi anvender for å representere satellittsystemet presenteres i dette kapitlet. Vi definerer notasjon og betegnelser, og gir en presis formulering av SS/TDMA trafikkoptimeringsproblemet med delproblemer.

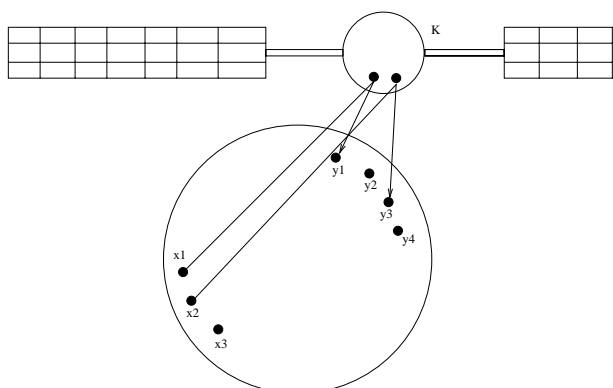
### 4.1 SS/TDMA systemets komponenter

Komponentene i et forenklet satellittkommunikasjonssystem som på figur 4.1 er  $m$  jordbaserte sendestasjoner  $x_1, \dots, x_m$  og  $n$  jordbaserte mottakerstasjoner  $y_1, \dots, y_n$ , samt en satellitt  $K$  i bane rundt jorden som skal formidle kommunikasjonen mellom dem. Satellitten er utstyrt med  $k$  antenner. Vi antar at antallet antenner er mindre eller lik antallet sendere og mottakere,  $k \leq \min(n, m)$ .

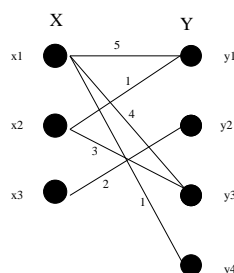
TDMA metoden løser som nevnt multiplexing problemet ved at antennene i løpet av korte tidsintervaller med faste svitsjtilstander formidler trafikk fra  $k$  sendere til  $k$  mottakere. Mellom disse tidsintervallene skjer svitsjingen der antennene skifter retning for å formidle trafikk fra  $k$  nye sendere til  $k$  nye mottakere. Det tar en viss tid å gjennomføre en svitsjing, *svitsjingtiden*  $t$ .

### 4.2 Trafikkbehovet

Vi kan representere trafikkbehovet i et SS/TDMA system med en  $m \times n$  ikke-negativ heltallsmatrise  $D = [d_{ij}]$ , der  $d_{ij}$  er trafikken som skal sendes fra  $x_i$  til  $y_j$  [5]. Trafikken  $d_{ij}$  måles i tidsenheten *slot time*, som er et mål på kanalkapasitet. Det tar én transponder  $d_{ij}$  tidsenheter å overføre den ønskede informasjonen  $x_i$  til  $y_j$ .  $D$  kalles *trafikkmatrisen*.



Figur 4.1: Forenklet satellittkommunikasjonssystem med  $m = 3$  sendestasjoner,  $n = 4$  mottakerstasjoner og  $k = 2$  transpondere. Her i en svitsjtilstand med trafikkformidling fra  $x_1$  til  $y_1$  og fra  $x_2$  til  $y_3$ .



Figur 4.2: Bipartitt graf representasjonen av en trafikkmatrise.

Trafikkbehovet kan også uttrykkes som en bipartitt graf  $G_D = (X, Y, K)$  med  $n + m$  noder [5]. Nodene i grafen er sender - og mottakerstasjonene  $x_1, \dots, x_m, y_1, \dots, y_n$ . Vi lar det gå en kant  $k_{ij}$  fra  $x_i$  til  $y_j$  dersom det skal gå trafikk mellom de to nodene. Kantens vekt er lik mengden trafikk målt i slot time. Hvis vi lar nodemengden  $X$  være de  $n$  sendernodene,  $Y$  være de  $m$  mottakernodene ser vi at grafen må være bipartitt.  $X$  og  $Y$  er disjunkte nodemengder siden ingen stasjon både kan være sender og mottaker, og alle kantene går fra noder i  $X$  til noder i  $Y$ , siden det ikke går trafikk mellom stasjoner av samme type.

Det eksisterer altså en 1-1 korrespondanse mellom den vektete bipartitte grafen  $G_D$  og trafikkmatrisen  $D$ . Det er like mange sendernoder i  $X$  som det er rader i  $D$  og like mange mottakernoder i  $Y$  som det er kolonner i  $D$ . En node  $x_i$  er forbundet med en node  $y_j$  ved en kant  $k_{ij}$  hvis og bare hvis  $d_{ij}$  er positiv. Antall kanter i grafen er dermed lik antall



positive elementer i trafikkmatrisen. Vekten til en kant  $k_{ij}$  er lik  $d_{ij}$ . Vi sier at  $D$  er grafens *reduerte nodeindikatormatrise*.

La oss se på et eksempel. Et lite satellittkommunikasjonssystem har  $m = 3$  sendestasjoner  $x_1, x_2, x_3$  og  $n = 4$  mottakerstasjoner  $y_1, y_2, y_3, y_4$ . I dette systemet skal det sendes trafikk som uttrykt i følgende trafikkmatrise:

$$D = \begin{bmatrix} 5 & 0 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

For en trafikkmatrise  $D$  vil vi her definere noen størrelser.

- $v(D)$  er det maksimale antallet positive elementer i en rad eller kolonne i  $D$ .
- $u(D)$  er den maksimale rad - eller kolonnesummen til  $D$ .
- $\rho(D)$  er antall positive elementer i  $D$ .
- $\sigma(D)$  er summen av elementene i  $D$ .

Vi kan også sette opp grafen  $G_D$ , se figur 4.2. Av både trafikkmatrisen og grafen kan vi nå lese trafikkbehovet til systemet. For eksempel ser vi at node  $x_1$  skal sende trafikk tilsvarende 4 enheter slot-time til node  $y_3$ . I trafikkmatrisen er elementet  $d_{13} = 4$ , og i grafen  $G_D$  forbindes  $x_1$  og  $y_3$  med en kant med vekt 4.

## 4.3 TDMA skjemaet

TDMA skjemaet inneholder informasjon om sekvensen av svitsjtilstander som utgjør en overføring av all trafikken derfinert av trafikkbehovet. For hver svitsjtilstand må systemet uttrykke hvilke sende - og mottakerstasjoner som kommuniserer, og hvor lenge svitsjtilstanden varer. Skjemaet kan presenteres på to forskjellige måter: som en sum av matriser og som en sekvens av grafer.

### 4.3.1 Matriserepresentasjon

Vi definerer først en *k-svitsjingmatrise* som en ikkenegativ heltallsmatrise med høyst ett positivt element i hver rad, høyst ett positivt element i hver kolonne og høyst  $k$  positive elementer til sammen.

I et SS/TDMA system med  $m$  sendernoder,  $n$  mottakernoder og  $k$  transpondere kan hver svitsjtilstand representeres med en slik  $m \times n$  *k-svitsjing* matrise  $S = [s_{ij}]$ . Hvert element  $s_{ij}$  i  $S$  er da lik trafikken som

overføres fra  $x_i$  til  $y_j$  i den aktuelle svitsjtilstanden. Siden vi kun har  $k$  transpondere kan vi selvsagt ha høyst  $k$  positive elementer, og da kan hver node bare sende til eller motta trafikk fra én annen node i hver svitsjtilstand, kan vi ha høyst ett positivt element i hver rad og i hver kolonne.

TDMA skjemaet kan dermed representeres som en sekvens av  $k$ -svitsjingmatriser  $S_1, S_2, \dots, S_s$ . En slik matriserepresentasjon har en viktig egenskap: Trafikkmatriksen  $D$  kan uttrykkes som summen av  $k$ -svitsjingmatrisene ([5]).

$$D = S_1 + S_2 + \dots + S_s$$

Problemet med å lage et TDMA skjema som gir informasjon om antall svitsjinger (som er lik antall svitsjtilstander) og svitsjtilstandenes varighet, kan altså reduseres til å finne en *dekomponering* av trafikkmatriksen  $D$  i en sum av  $k$ -svitsjing matriser.

*Overføringstiden* til en svitsjtilstand er lik den største mengden trafikk som skal overføres i løpet av tilstanden. Hvis  $S_l = [s_{ij}]$  er  $k$ -svitsjingmatrisen til svitsjtilstand  $l$  vil da overføringstiden  $T(S_l)$  til svitsjtilstanden være gitt ved

$$T(S_l) = \max_{ij} s_{ij}$$

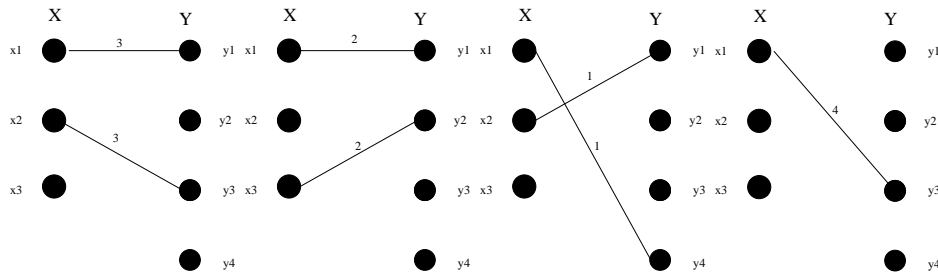
Den totale overføringstiden  $t$  til TDMA skjemaet  $S_1, S_2, \dots, S_s$  defineres som summen av overføringstidene til de enkelte  $k$ -svitsjingmatrisene. Tiden det tar å foreta hver svitsjing er ikke tatt med.

$$T = \sum_{l=1}^s T(S_l)$$

For å få en bedre forståelse kan vi gå tilbake til eksemplet fra avsnitt 4.2. Vi vil anta at satellitten har  $k = 2$  transpondere. En mulig TDMA frame for vårt eksempel med  $s = 4$  svitsjtilstander med varighet hhv. 3, 2, 1 og 4 kan representeres slik:

$$\begin{aligned} D &= \begin{bmatrix} 5 & 0 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Med dette skjemaet blir den totale overføringstiden  $T_1 = 3 + 2 + 1 + 4 = 10$ .



Figur 4.3: Matchinger som representasjon av TDMA skjema,  $s = 4$  svitsjinger.

### 4.3.2 Grafrepresentasjon

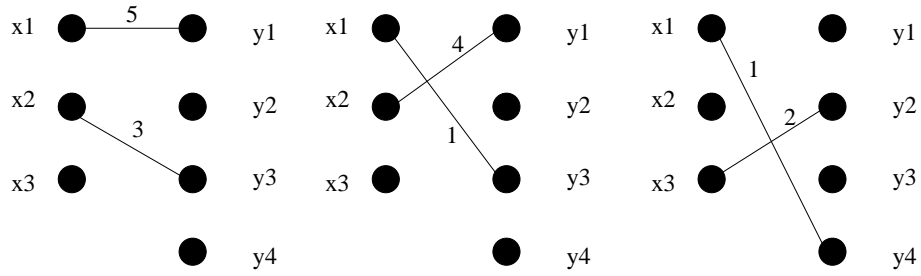
Hver av svitsjtilstandene i et SS/TDMA skjema kan også representeres ved en matching med høyst  $k$  kanter i den bipartitte grafen  $G_D$ . En kant fra  $x_i$  til  $y_j$  betyr at det i svitsjtilstanden går trafikk fra sendestasjon  $x_i$  til mottakerstasjon  $y_j$ . Kantens vekt er trafikkmengden (målt i slot time), som overføres. Den største vekten til en kant i matchingen er lik svitsjtilstandens varighet. Også her har vi en 1-1 korrespondanse: mellom  $k$ -svitsjingmatrisen og matchingen, som begge representerer samme svitsjtilstand.  $k$ -svitsjingmatrisene er reduserte indikatormatriser til matchingene.

I figur 4.3 ser vi TDMA skjemaet i eksempelet ovenfor representert ved en følge av 4 matchinger. Disse matchingene korresponderer med de 4  $k$ -svitsjingmatrisene fra dekomponeringen av trafikkmatriksen.

## 4.4 Preemption

Hvorvidt det benyttes preemptiv eller ikke-preemptiv skedulering vil gjenspeiles i den matematiske modellen av TDMA skjemaet. I det preemptive tilfellet tillates som nevnt at parene av sender - og mottakerstasjoner ikke har overført all trafikken før de overlater bruken av antennen til neste par, for så etter et viss antall svitsjinger å få overføre trafikk igjen. I modellen tilsvarer dette at et element  $d_{ij}$  i trafikkmatriksen  $D$  kan deles opp  $s_{lij}$ ,  $l = 1, \dots, s$ , som er elementene i  $k$ -svitsjingmatrisene  $S_l$ ,  $l = 1, \dots, s$ , slik at  $d_{ij} = s_{1ij} + s_{2ij} + \dots + s_{sij}$ .

Ved ikke-preemptiv skedulering må all trafikken til et par overføres i løpet av én svitsjtilstand. Dette vil tilsvare at for hvert element  $d_{ij}$  i trafikkmatriksen finnes det kun én  $k$ -svitsjingmatrise  $S_l$  med positivt element  $s_{lij}$ . Vi har da at  $d_{ij} = s_{lij}$  for en  $l \in \{1, 2, \dots, s\}$ .



Figur 4.4: Matchinger som representasjon av TDMA skjema,  $s = 3$  svitsjinger.

#### 4.5 SS/TDMA trafikkoptimeringsproblemet og varianter

Som nevnt tar det en viss tid å rekonfigurere antennenenes posisjon mellom hver svitsjtilstand. Antall svitsjinger er derfor viktig for den totale tiden det tar å formidle all trafikken gjennom systemet. TDMA skjemaet vi konstruerte i eksemplet vårt over, med  $s = 4$  svitsjinger, var bare et av mange gyldige skjemaer. Her et annet, denne gangen med  $s = 3$  svitsjinger (grafrepresentasjonen, se figur 4.4):

$$D = \begin{bmatrix} 5 & 0 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 4 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

Overføringstiden for dette skjemaet er  $T_2 = 5 + 4 + 2 = 11$ , mot  $T_1 = 10$  for det forrige. Overføringstiden er altså høyere, men antall svitsjinger lavere. Hvilket som er det beste skjemaet avhenger dermed av hvor lang tid hver svitsjing tar.

Vi definerer *varigheten*  $L$  til en TDMA frame ved

$$L = T + st$$

$T$  er overføringstiden,  $s$  er antall svitsjinger og  $t$  er svitsjingtiden.

I vårt eksempel blir varigheten av de to TDMA skjemaene vi har sett på hhv.  $L_1 = 10 + 4d$  og  $L_2 = 11 + 3d$ . Vi ser at  $L_1 < L_2$  hvis og bare hvis

$t < 1$ .

SS/TDMA trafikkoptimeringsproblemet kan da uttrykkes slik:

**Problem 1. TOP-L.**

*I et SS/TDMA satellittsystem med  $k$  antenner og svitsjingtid  $t$ , gitt et trafikkbehov definert av trafikkmatrisen  $D$ , finn et TDMA skjema bestående av en sekvens av  $k$ -svitsjingmatriser  $S_1, S_2, \dots, S_s$  med korresponderende overføringstider  $T(S_1), T(S_2), \dots, T(S_s)$ , slik at varigheten av skjemaet*

$$L = T + st = \sum_{l=1}^s T(S_l) + st$$

*minimeres.*

Dette er et komplekst problem som involverer de to parameterne  $T$  og  $s$ . Samtidig som vi ønsker å finne minimal overføringstid  $T$  ønsker vi å finne en optimal dekomponering av trafikk matrisen  $D$  i færrest mulig  $k$ -svitsjingmatriser. Som vi ser i eksemplet øker gjerne den ene når den andre avtar og omvendt. Løsningen av optimeringsproblemet må balansere ønskene om liten overføringstid og få  $k$ -svitsjingmatriser slik at varigheten  $L$  minimeres.

I vår tilnærming til problemet med å minimere den totale varigheten, vil det være nyttig å se på minimering av overføringstiden  $T$  og antallet svitsjinger  $s$  hver for seg. Vi definerer dermed også de to følgende delproblemer:

**Problem 2. TOP-T.**

*I et SS/TDMA satellittsystem med  $k$  antenner, gitt et trafikkbehov definert av trafikkmatrisen  $D$ , finn et TDMA skjema bestående av en sekvens av  $k$ -svitsjingmatriser  $S_1, S_2, \dots, S_s$  med korresponderende overføringstider  $T(S_1), T(S_2), \dots, T(S_s)$ , slik at overføringstiden til skjemaet*

$$T = \sum_{l=1}^s T(S_l)$$

*minimeres.*

**Problem 3. TOP-s.**

*I et SS/TDMA satellittsystem med  $k$  antenner, gitt et trafikkbehov definert av trafikkmatrisen  $D$ , finn et TDMA skjema bestående av en sekvens av  $k$ -svitsjingmatriser  $S_1, S_2, \dots, S_s$  med minimum antall svitsjinger.*

I alle de tre foregående problemene antar vi at det benyttes preemptiv skedulering. Ved å i stedet anta ikke-preemptiv skedulering, men ellers likelydende formulering, får vi problemene *TOP-L-ip*, *TOP-T-ip* og *TOP-s-ip*. Flere resultater er identiske for begge klassene av problemer, men vi skal i analysen i kap 5 se på noen egenskaper som skiller det ikke-preemptive fra det preemptive SS/TDMA trafikkoptimeringsproblemet.

Flere artikler om SS/TDMA trafikkoptimeringsproblemet forenkler det originale problemet noe, og antar at det er akkurat like mange antenner som det minste antallet sender - eller mottakerstasjoner. Noen av resultatene som presenteres gjelder kun for dette spesialtilfellet, og vi definerer derfor enda en klasse med varianter av problemene. I problemene *TOP-L-k*, *TOP-T-k* og *TOP-s-k* antas det at  $k = \min(n, m)$ , men de er ellers likelydende med TOP-L, TOP-T og TOP-s. For tilfellene hvor  $k = \min(n, m)$  og det benyttes ikke-preemptiv skedulering definerer vi likeledes problemene *TOP-L-k-ip*, *TOP-T-k-ip* og *TOP-s-k-ip*.

## Kapittel 5

# Analyse

I analysekapittelet presenterer vi først en rekke kjente resultater om skranker og optimale løsninger av problemet. Vi foretar så en kompleksitetsanalyse av problemet og viser at TOP-L-k er *NP*-komplett. For visse trafikkmatriser lar problemet seg dog løse polynomt. Vi kan generalisere disse matrisene, og definerer en klasse trafikkmatriser som gir en polynom løsning av TOP-L. Vi viser også et kompleksitetsresultat for tilnærming av TOP-L-k.

### 5.1 Skranker og minimumsverdier

Et første steg mot løsning av SS/TDMA trafikkoptimeringsproblemet TOP-L er å se på noen resultater om de to delproblemene TOP-s og TOP-T. Fra “Minimizing the number of switchings in an SS/TDMA system” [10] av Gopal og Wong og “An optimum slot assignment algorithm for an SS/TDMA system with variable number of transponders” [4] av Bongiovanni, Coppersmith og Wong fås fundamentale teoremer som gir oss minimale verdier av overføringstiden  $T$  og antall svitsjinger  $s$ . Begge de to delproblemene har altså en løsning.

Teoremene finner minimale verdier av parameterne  $T$  og  $s$  uavhengig av hverandre. Minimumsverdien av  $s$  er funnet uten tanke for overføringstiden  $T$  i et slikt skjema, og minimumsverdien av  $T$  tar ikke hensyn til antall svitsjinger  $s$ .

Vi ser på løsningene i to deler. Først presenteres teoremer som gir nedre skranker for  $T$  og  $s$ . Dette er forholdsvis enkle bevis som kun tar utgangspunkt i lett forståelige egenskaper ved trafikkmatrisen. Deretter legges det fram teoremer som sier at disse nedre skrankene er oppnåelige og dermed minimumsverdier. I kapittel 6 gis algoritmer som gir TDMA skjemaer med disse minimumsverdiene av overføringstid og antall

svitsjinger, og dermed har vi konstruksjonsbevis for begge teoremene.

I spesialtilfellet med  $k = \min(m, n)$  kan resultatene forenkles noe, og vi presenterer derfor også korollarer som gir optimale løsninger av TOP-s-k og TOP-T-k.

### 5.1.1 Nedre skranke for s i TOP-s

For en ikke-negativ matrise  $X$ , la  $\rho(X)$  være antall positive elementer i  $X$  og la  $\nu(X)$  være det maksimale antall positive elementer i en rad eller kolonne i  $X$ . Fra [10] har vi da følgende teorem:

**Teorem 3.** *Gitt en trafikkmatrise  $D$  er*

$$\max\{\nu(D), \lceil \frac{\rho(D)}{k} \rceil\}$$

*en nedre skranke for antall  $k$ -svitsjinger i en TDMA frame.*

**Bevis:** I en trafikkmatrise  $D = [d_{ij}]$  er antallet positive elementer i en rad  $i$  lik antallet mottakerstasjoner som skal motta signaler fra  $x_i$ , dvs like mange ganger som satelitten må svitsje for å overføre all trafikken fra  $x_i$  (husk at vi regner én svitsjing ved systemets første konfigurasjon av antennene). Antallet positive elementer i en kolonne  $j$  er lik antallet sendestasjoner som skal sende trafikk til  $y_j$ . For at all trafikken skal formidles må således antall  $k$ -svitsjinger være minst like stort som det største antallet overføringer fra eller til en stasjon, dvs det maksimale antallet positive elementer i en rad eller kolonne i  $D$ .

Samtidig kan vi se på  $\rho(D)$ , antallet positive elementer i  $D$ .  $\rho(D)$  er lik mengden av koblinger som skal gjøres mellom sende - og mottakerstasjoner. Siden vi har  $k$  transpondere å foreta disse koblingene gjennom, må vi svitsje minst  $\lceil \frac{\rho(D)}{k} \rceil$  ganger.

Siden skjemaet vårt må overholde kravene fra begge de foregående avsnittene, har vi altså som i teoremet at antall svitsjinger må være minst lik det største av det maksimale antall positive elementer i en rad eller kolonne i  $D$  og  $\lceil \frac{\rho(D)}{k} \rceil$ . □

### 5.1.2 Nedre skranke for s i TOP-s-k

I tilfellet der antallet antenner er lik antall sender - eller mottakerstasjoner, dvs  $k = \min(m, n)$ , kan vi forenkle resultatet i teorem 3 noe.



**Korollar 4.** Gitt en trafikkmatrise  $D$  i et SS/TDMA satelittsystem med  $n$  senderstasjoner,  $m$  mottakerstasjoner og  $k = \min(m, n)$  antenner, er

$$\nu(D)$$

en nedre skranke for antall  $k$ -svitsjinger i en TDMA frame.

**Bevis:** I en  $m \times n$  matrise  $D$  med  $\rho(D)$  positive elementer og  $k = \min(m, n)$ , må det finnes minst  $\lceil \frac{\rho(D)}{k} \rceil$  positive elementer i én rad eller kolonne i  $D$ , dvs.  $\nu(D) \geq \lceil \frac{\rho(D)}{k} \rceil$ . Det følger da av teorem 3 at  $\nu(D)$  er en nedre skranke for antall  $k$ -svitsjinger.  $\square$

### 5.1.3 Nedre skranke for T i TOP-T

For en ikke-negativ matrise  $X$ , la  $\sigma(X)$  være summen av elementene i  $X$  og  $u(X)$  den maksimale rad- eller kolonnesummen til  $X$ . Fra [4] har vi følgende teorem:

**Teorem 4.** Gitt en trafikkmatrise  $D$  er

$$\max\{u(D), \lceil \frac{\sigma(D)}{k} \rceil\}$$

en nedre skranke for overføringstiden til en TDMA frame.

**Bevis:** Radsummen til en rad  $i$  i en trafikkmatrise  $D = [d_{ij}]$  er mengden trafikk (målt i slot time) som skal overføres fra  $x_i$  til  $y_1, \dots, y_n$ . Kolonnesummen til en kolonne  $j$  i  $D$  er mengden trafikk som skal overføres fra  $x_1, \dots, x_m$  til  $y_j$ . Siden all trafikk skal overføres i løpet av TDMA skjemaet, må overføringstiden  $T$  være minst like stor som den største av trafikkoverføringene til eller fra en stasjon, dvs minst like stor som den maksimale rad - eller kolonnesummen til  $D$ .

Samtidig kan vi se på  $\sigma(D)$ , summen av elementene i trafikkmatrisen.  $\sigma(D)$  er lik mengden av all trafikk (målt i slot time) som skal sendes i systemet. Siden vi har  $k$  transpondere som formidler trafikk samtidig, kan ikke overføringstiden være mindre enn  $\lceil \frac{\sigma(D)}{k} \rceil$ .

Et TDMA skjema må overholde begge de foregående kravene, altså kan ikke overføringstiden være mindre enn det største av den maksimale rad - eller kolonnesummen til  $D$  og  $\lceil \frac{\sigma(D)}{k} \rceil$ .  $\square$

### 5.1.4 Nedre skranke for T i TOP-T-k

Som for teorem 3 kan vi i tilfellet  $k = \min(m, n)$  forenkle også resultatet fra teorem 4.

**Korollar 5.** *Gitt en  $m \times n$  trafikkmatrise  $D$  i et SS/TDMA satelittsystem med  $k = \min(m, n)$  antenner, er*

$$u(D)$$

*en nedre skranke for overføringstiden til en TDMA frame.*

**Bevis:** I en  $m \times n$  matrise  $D$  der summen av elementene er lik  $\sigma(D)$  og  $k = \min(m, n)$ , må det finnes en rad - eller kolonnesum minst lik  $\lceil \frac{\sigma(D)}{k} \rceil$ , dvs.  $u(D) \geq \lceil \frac{\sigma(D)}{k} \rceil$ . Det følger da av teorem 4 at  $u(D)$  er en nedre skranke for overføringstiden.  $\square$

### 5.1.5 Minimumsverdi av s i TOP-s

Som vi skal se kan vi lett finne et enda sterkere resultat enn teorem 3 ved å vise at den nedre skranken for antall svitsjinger i en TDMA frame er oppnåelig, og dermed en minimumsverdi av  $s$ . Fra [10] har vi nemlig:

**Teorem 5.** *Gitt en trafikkmatrise  $D$  i et SS/TDMA satelittsystem med  $k$  antenner, er det minimale antallet  $k$ -svitsjinger  $s_0$  i en TDMA frame lik*

$$\max\{\nu(D), \lceil \frac{\rho(D)}{k} \rceil\}$$

**Bevis:** Fra teorem 3 har vi at  $\max\{\nu(D), \lceil \frac{\rho(D)}{k} \rceil\}$  er en nedre skranke for antall svitsjinger. For bevis for at denne nedre skranken faktisk er oppnåelig og dermed minimumsverdien av  $s$ , henvises det til kapittel 6, der algoritme A-TOP-s gir et slikt TDMA skjema med antall svitsjinger lik den nedre skranken.  $\square$

I trafikkmatrisen  $D$  fra vårt eksempel i avsnitt 4.2 er det maksimale antall positive elementer i en rad eller kolonne lik 3, nemlig i matrisens første rad. Det skal trafikk fra  $x_1$  til  $y_1$ ,  $y_3$  og  $y_4$ . Satelitten må altså svitsje minst 3 ganger. Antall positive elementer i  $D$  er  $\rho(D) = 6$ . Det skal gjøres 6 koblinger, og med  $k = 2$  transpondere må vi svitsje minst  $\lceil \frac{\rho(D)}{k} \rceil = \frac{6}{2} = 3$ . Teorem 1 gir at det minimale antall  $k$ -svitsjinger er det største av disse tallene. I vårt eksempel er de tilfeldigvis like,  $s_0 = 3$ . Altså hadde vi i avsnitt 4.5 funnet et skjema med minimalt antall svitsjinger.

### 5.1.6 Minimumsverdi av $s$ i TOP-s-k

Igjen kan vi i tilfellet  $k = \min(m, n)$  forenkle resultatet noe.

**Korollar 6.** *Gitt en  $m \times n$  trafikkmatrise  $D$  i et SS/TDMA satellittsystem med  $k = \min(m, n)$  antenner, er det minimale antallet  $k$ -svitsjinger  $s_0$  i en TDMA frame lik*

$$v(D)$$

**Bevis:** Følger av teorem 5 og korollar 4. □

### 5.1.7 Minimumsverdi av $T$ i TOP-T

På samme måte som for antall svitsjinger kan vi vise at også den nedre skranken for overføringstiden er oppnåelig, og dermed en minimumsverdi av  $T$ . Fra [4] har vi:

**Teorem 6.** *Gitt en trafikkmatrise  $D$  i et SS/TDMA satellittsystem med  $k$  antenner, er den minimale overføringstiden  $T_0$  i en TDMA frame lik*

$$\max\{u(D), \lceil \frac{\sigma(D)}{k} \rceil\}$$

**Bevis:** Fra teorem 4 har vi at  $\max\{u(D), \lceil \frac{\sigma(D)}{k} \rceil\}$  er en nedre skranke for overføringstiden. I kapittel 6 beskrives algoritmen A-TOP-T, som gir et TDMA skjema med overføringstid lik den nedre skranken. Dette fullfører beviset. □

Den maksimale rad - eller kolonnesum til  $D$  fra eksemplet i avsnitt 4.2 er summen av elementene i den første raden:  $\sum_{j=1}^4 d_{1j} = 10$ . Summen av alle elementene i  $D$  blir  $\sigma(D) = 16$ , slik at  $\lceil \frac{\sigma(D)}{k} \rceil = \lceil \frac{16}{2} \rceil = 8$ . Teoremet gir da at den minimale overføringstiden er lik det største av  $\sum_{j=1}^4 d_{1j}$  og  $\lceil \frac{\sigma(D)}{k} \rceil$ , dvs  $T_0 = 10$ . Altså oppnådde vi i vårt første skjema fra avsnitt 4.3.1 minimal overføringstid for  $D$ .

### 5.1.8 Minimumsverdi av $T$ i TOP-T-k

I tilfellet  $k = \min(m, n)$  kan vi igjen forenkle resultatet.

**Korollar 7.** *Gitt en  $m \times n$  trafikkmatrise  $D$  i et SS/TDMA satellittsystem med  $k = \min(m, n)$  antenner, er den minimale overføringstiden  $T$  i en TDMA frame lik*

$$u(D)$$

**Bevis:** Følger av teorem 6 og korollar 5. □

### 5.1.9 Nedre skranker for $L$ i TOP-L og TOP-L-k

Etter å ha funnet nedre skranker for overføringstiden  $T$  og antall svitsjinger  $s$  i en TDMA frame, følger en nedre skranke for varigheten  $L = T + st$  umiddelbart. De to neste korollarene gir nedre skranker for  $L$  i problemene TOP-L og TOP-L-k.

**Korollar 8.** *Gitt en trafikkmatrise  $D$  i et SS/TDMA satellittsystem med  $k$  antenner er*

$$L_0 = \max\{u(D), \lceil \frac{\sigma(D)}{k} \rceil\} + \max\{v(D), \lceil \frac{\rho(D)}{k} \rceil\}t$$

*en nedre skranke for varigheten  $L$  av en TDMA frame.*

**Bevis:** Følger av teorem 3 og teorem 4. □

**Korollar 9.** *Gitt en  $m \times n$  trafikkmatrise  $D$  i et SS/TDMA satellittsystem med  $k = \min(m, n)$  antenner, er*

$$L_0 = u(D) + v(D)t$$

*en nedre skranke for varigheten  $L$  av en TDMA frame.*

**Bevis:** Følger av korollar 4 og korollar 5. □

Siden  $s_0 = 3$  og  $T_0 = 10$  er nedre skranker for antall  $k$ -svitsjingmatriser og overføringstid i eksemplet vårt, vil  $L_0 = T_0 + s_0t = 10 + 3t$  være en nedre skranke for varigheten av en TDMA frame for  $D$ . Men selv om de første skrankene lot seg oppnå, kan vi likevel ikke utfra dette svare på om  $L_0 = 10 + 3t$  faktisk kan *oppnås* for et TDMA skjema. Vi fikk nemlig  $s = s_0 = 3$  for et skjema der  $T = 11$ , og  $T = T_0 = 10$  for et skjema der  $s = 4$ . Teoremene sier ikke noe om at de nedre skrankene for de to parameterne *ikke* kan oppnås for ett og samme TDMA skjema, men som vi aner fra eksempelet virker det sannsynlig at i de fleste (tilstrekkelig store) systemer vil et lavt antall  $s$   $k$ -svitsjinger medføre en høy overføringstid  $T$  og omvendt vil en lav  $T$  medføre en stor  $s$ .

I kompleksitetsanalysen vil vi vise at SS/TDMA trafikkoptimeringsproblemet TOP-L-k faktisk er NP-komplett, og vi må derfor nøye oss med å finne gode tilnærminger til en optimal løsning. Vurderingen av hvor gode disse tilnæringene er, blir gjort opp mot de nedre skrankene for  $L$  som gitt i korollar 8 og 9.

For  $k = 1$  har vi et spesielt tilfelle. I et TDMA skjema som formidler all trafikk i et TDMA satellittsystem over én antenne, vil hver k-svitsjingmatrise inneholde akkurat ett av elementene i trafikkmatrisen  $D$ . Skjemaet vil da inneholde  $\lceil \frac{\rho(D)}{k} \rceil = \rho(D)$  svitsjinger, og overførings-tiden vil bli lik  $\lceil \frac{\sigma(D)}{k} \rceil = \sigma(D)$ . Vi oppnår da en varighet  $L$  lik den nedre skranken  $L_0$ . Begge algoritmene A-TOP-s og A-TOP-L i kapittel 6 vil produsere TDMA skjema med optimal løsning av TOP-L.

### 5.1.10 Skranker ved ikke-preemptiv skedulering

Bevisene for teorem 3 og korollar 4 som gir nedre skranker for antall svitsjinger  $s$  i hhv. TOP-s og TOP-s-k er ikke avhengig av preemptiv skedulering. Teoremene gjelder derfor også i de ikke-preemptive tilfellene TOP-s-ip og TOP-L-ip. For TDMA skjemaer med ikke-preemptiv skedulering har vi i tillegg en øvre skranke på antallet svitsjinger, nemlig antallet positive elementer i trafikkmatrisen. Siden hvert par av sender - og mot-takerstasjoner gjør seg ferdig før antennen overlates til neste par, vil et TDMA skjema aldri behøve flere svitsjinger enn det finnes antall trafikk-overføringer mellom distinkte par av stasjoner i et trafikkbehov, altså positive elementer i trafikkmatrisen. Gitt at vi ikke tillater tomme svitsjinger i en TDMA frame, kan vi dermed formulere følgende preposisjon:

**Preposisjon 5.1.** *Gitt et trafikkmatrise  $D$  er antallet positive elementer  $\rho(D)$  i trafikkmatrisen en øvre skranke for antallet svitsjinger i en TDMA frame uten preemptiv skedulering.*

Dermed gjelder følgende ulikhet for antaller svitsjinger  $s$  i problemet TOP-s-ip med trafikkmatrisen  $D$ :

$$\max\{\nu(D), \lceil \frac{\rho(D)}{k} \rceil\} \leq s \leq \rho(D)$$

I tilfellet  $k = \min(m, n)$  (problem TOP-s-k-ip) kan som følge av korollar 4 ulikheten forenkles til  $\nu(D) \leq s \leq \rho(D)$ .

## 5.2 Kompleksitetsanalyse

Vi har i forrige delkapittel presentert nedre skranker for  $s$ ,  $T$  og  $L$ , og i kapittel 6 gis algoritmene som i polynom tid løser problem [?] og [?] ved å produsere TDMA skjemaer med minimale verdier av henholdsvis  $s$  og  $T$ . Vi ønsker å finne en polynom algoritme som løser problem TOP-L

og finner minimal verdi for  $L$ . Kompleksitetsanalysen viser at vil letingen etter en slik algoritme med all sannsynlighet være nytteløst for  $k = \min(m, n)$ , da TOP-L-k er NP-komplett.

### 5.2.1 NP-komplettethet av TOP-L-k og TOP-L

Det viser seg at kjerneproblemet i oppgaven, SS/TDMA trafikkoptimeringsproblemet TOP-L, er NP-komplett [4]. Dette bevises ved en reduksjon fra et kjent NP-komplett problem, "The Timetable Design Problem" (TTD) [9].

**Problem 4.** "The Timetable Design Problem" (TTD).

Gitt:

- 1) En mengde  $H = \{h_1, h_2, \dots, h_l\}$  av  $l$  timer
- 2) En mengde  $F = \{F_1, F_2, \dots, F_n\}$  av  $n$  lærere
- 3) En mengde  $G = \{G_1, G_2, \dots, G_m\}$  av  $m$  klasser
- 4) En relasjon  $A : F \rightarrow H$  som representerer tilgjengeligheten av lærere, der lærer  $T_i$  er tilgjengelig i time  $h_k$  hvis og bare hvis  $h_k \in A(F_i)$
- 5) En relasjon  $B : G \rightarrow H$  som tilsvarende representerer tilgjengeligheten av klasser
- 6) Et  $n \times m$  kravmatrise  $R$ , der hvert element  $r_{ij}$  representerer kravet at lærer  $F_i$  må undervise klasse  $G_j$  i  $r_{ij}$  timer.

*Problem:* Finnes det en tilordningsfunksjon  $f : F \times G \times H \rightarrow \{0, 1\}$  slik at følgende holder?

- 1)  $f(F_i, G_j, h_k) = 1 \Rightarrow h_k \in A(F_i) \cap B(G_j)$ , dvs. lærer og klasse må være tilgjengelige
- 2)  $\sum_{F_i \in F} f(F_i, G_j, h_k) \leq 1$  for alle  $G_j \in G, h_k \in H$ , dvs. ingen klasse undervises av mer en en lærer i en time
- 3)  $\sum_{G_j \in G} f(F_i, G_j, h_k) \leq 1$  for alle  $F_i \in F, h_k \in H$ , dvs. ingen lærer underviser i mer enn en klasse i en time
- 4)  $\sum_{h_k \in H} f(F_i, G_j, h_k) = R_{ij}$  for alle  $F_i \in F, G_j \in G$ , dvs. kravmatrisen er tilfredsstilt

Vi skal ta utgangspunkt i et spesialtilfelle av dette problemet, "the Reduced Timetable Design Problem" [7], RTTD, med bare 3 timer ( $l = 3$ ), der klasser alltid er tilgjengelige ( $B(G_i) = H \forall G_i \in G$ ). Hver lærer har kun en klasse i én time, altså er kravmatrisen  $R$  en 0-1 matrise.

**Problem 5.** "The Reduced Timetable Design Problem" (RTTD).

Gitt:

- 1) En mengde  $F = \{F_1, F_2, \dots, F_n\}$  av  $n$  lærere
- 2) En mengde  $G = \{G_1, G_2, \dots, G_m\}$  av  $m$  klasser

3) En tilgjengelighetsfunksjon  $A : F \times \{1, 2, 3\} \rightarrow \{0, 1\}$

4) En  $\{0, 1\}$  kravmatrise  $R$

*Problem:* Finnes det en tilordningsfunksjon  $f : F \times G \times \{1, 2, 3\} \rightarrow \{0, 1\}$  slik at følgende holder?

1)  $f(F_i, G_j, k) = 1 \Rightarrow A(F_i, k) = 1$

2)  $\sum_{i=1}^n f(F_i, G_j, h) \leq 1$  for alle  $G_j$  og  $h \in \{1, 2, 3\}$

3)  $\sum_{j=1}^m f(F_i, G_j, h) \leq 1$  for alle  $F_i$  og  $h \in \{1, 2, 3\}$

4)  $\sum_{h=1}^3 f(F_i, G_j, h) = R_{ij}$  for alle  $F_i$  og  $G_j$

Fra [9] har vi følgende resultat:

**Teorem 7.** *RTTD er NP-komplett.*

Vi viser så at SS/TDMA trafikkoptimeringsproblemet TOP-L-k kan reduseres til RDDT, og dermed har vi bevist at TOP-L-k også er NP-komplett. Beviset er hentet fra fra [4].

**Teorem 8.**  $TOP-L-k \propto RTTD$ .

**Bevis:** For å bevise at SS/TDMA trafikkoptimeringsproblemet lar seg redusere til “The Reduced Timetable Design Problem” definerer vi en instans av TOP-L-k gitt en instans av RTTD, og viser så at instansen av av TOP-L-k har en løsning hvis og bare hvis det finnes en løsning av instansen av RTTD.

Gitt en instans  $\langle F, G, A, R \rangle$  av RTTD. Vi konstruerer da en  $N \times M$  trafikkmatrise  $D$ , der  $N = 2n$  og  $M = n + m$ , på følgende måte:

1. La den øvre venstre  $n \times m$  submatrisen være lik 0-1 kravmatrisen  $R$  fra RTTD.
2. Fyll den nedre venstre  $n \times m$  submatrisen med 0-elementer.
3. Den høyre  $2n \times n$  submatrisen konstrueres fra tilgjengelighetsfunksjonen  $A$  på følgende måte:
  - For  $1 \leq i \leq n$  sett  $d_{ij} = h$  for alle  $(i, j)$  der  $j = m + 1 + [(i + h - 2) \bmod n]$  og  $A(F_i, h) = 0$  for en  $h \in \{1, 2, 3\}$ .
  - For  $n + 1 \leq i \leq 2n$  sett  $d_{ij} = h$  for alle  $(i, j)$  der  $j = m + 1 + [(i - n + h - 2) \bmod n]$  og  $A(F_{i-n}, h) = 1$  for en  $h \in \{1, 2, 3\}$ .

- Alle andre  $d_{ij}$  settes lik 0.

Som et eksempel skal vi se på følgende instans av RTTD. Vi lar  $|F| = n = 3$  og  $|G| = m = 4$ . Matrisen  $A = [a_{ij}]$  gir tilordningsfunksjonen ( $a_{ij} = A(F_i, j)$ ) og  $R$  er kravmatrisen:

$$R = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Dette gir følgende  $6 \times 7$  trafikkmatrise  $D$ , som definerer den korresponderende instansen av TOP-L:

$$D = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 2 & 3 \\ 0 & 1 & 0 & 1 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 0 \end{bmatrix}$$

På bakgrunnen av den konstituerte trafikkmatrisen kan vi så definere følgende instans av TOP-L-k: *Gitt trafikkmatrisen  $D$ , finn et TDMA skjema med minimum antall svitsjinger  $s_0$  og overføringstid  $T = 6$ .* Vi skal så bevise at dette problemet har en løsning hvis og bare hvis den opprinnelige instansen av RTTD har en løsning.

Før dette beviset skal vi se at det minimale antallet svitsjinger er lik 3. Siden det kun er 3 timer i problemet RTTD kan ikke  $R$  ha mer enn 3 positive elementer i en kolonne. Det samme gjelder derfor for kolonnene  $1, \dots, m$  i trafikkmatrisen  $D$ . Konstruksjonen av  $D$  gir hver av kolonnene  $m + 1, \dots, n + m$  3 positive elementer. Likeledes sørger konstruksjonen av  $D$  for at radene  $n + 1, \dots, 2n$  hver har tre positive elementer. Antall positive elementer i rad  $i$  for  $i = 1, \dots, n$  er lik antall timer lærer  $F_i$  underviser samt antall timer læreren ikke er tilgjengelig ( $A(F_i, h) = 0$  for en  $h \in \{1, 2, 3\}$ ). Dermed er det maksimale antallet positive elementer i en rad eller kolonne i  $D$  lik 3, dvs  $\nu(D) = 3$ . Det følger da av korollar 6 at det minimale antallet svitsjinger i en TDMA frame er lik 3.

*Hvis:* Vi antar vi har en løsning av RTTD. Fra denne løsningen, velg elementene i kravmatrisen  $R$  som tilsvarende undervisning i time 1, og de tilsvarende i trafikkmatrisen  $D$ . Ved krav 2 og 3 i problemet er vi sikret at ingen av disse elementene ligger på samme rad eller kolonne. Vi velger så ut elementene med verdi lik 1 fra den høyre  $2n \times n$  submatrisen av  $D$ . Konstruksjonen av  $D$  sikrer at ingen av disse ligger på samme rad eller kolonne. Videre vil ingen av disse elementene ligge på samme rad



som en av elementene fra  $R$  som tilsvarer undervisning i time 1. Hvis dette siste ikke var tilfelle ville det implisere at en lærer som underviste i time 1 ikke var tilgjengelig i time 1. Alle disse elementene utgjør derfor en k-svitsjningmatrise der alle elementer lik 1.

På tilsvarende måte finner vi k-svitsjningmatriser med elementer lik 2 og 3. Disse utgjør tilsammen en TDMA frame med minimalt antall svitsjninger og overføringsstid  $T = 1 + 2 + 3 = 6$ . Vi har derfor en løsning av instansen av problemet TOP-L.

*Bare hvis:* Anta en instans av et TOP-L-k gitt trafikkmatrisen  $D$  konstruert som over. Problemet har en løsning med minimalt antall svitsjninger  $s_0 = 3$  og overføringstid  $T = 6$ . Fra den høyre  $2n \times n$  submatrisen til  $D$ , som vi kaller  $E$ , er det maksimalt  $n$  elementer som kan være inneholdt i en k-svitsjningmatrise. Det er 3 svitsjninger i en løsning av TOP-L-k. Fra konstruksjonen inneholder  $E$   $n$  elementer med hver av verdiene 1, 2, 3. Den eneste måten som en overføringstid  $T = 6$  kan oppnås på er at alle  $n$  elementer med verdi 3 er i én k-svitsjningmatrise, alle  $n$  elementene med verdi 2 i en annen og alle  $n$  med verdi 1 i den tredje.

I k-svitsjningmatrisen med elementene fra  $E$  med verdi 1,  $S_1$  ser vi så på den øvre venstre  $n \times m$  submatrisen som korresponderer med kravmatrisen  $R$ . Fra k-svitsjningmatrisens egenskaper vet vi at ingen av elementene i denne submatrisen kan ligge på samme rad eller kolonne. Dermed vil disse elementene kunne representere undervisningen i time 1, ved at  $S_{1ij} = f(i, G_j, 1)$ .

Har da at følgende er oppfylt for  $h = 1$ :

- 1)  $f(F_i, G_j, 1) = 1 \Rightarrow A(F_i, h) = 1$  (Hvis  $A(F_i) = 0$  ville vi hatt et element med verdi 1 i rad  $i$ .)
- 2)  $\sum_{i=1}^n f(F_i, G_j, h) \leq 1$  for alle  $G_j$
- 3)  $\sum_{j=1}^m f(F_i, G_j, h) \leq 1$  for alle  $F_i$

På samme måte vil k-svitsjningmatrisene  $S_2$  som inneholder elementene med verdi lik 2 representere undervisningen i time 2. Tilsvarende for k-svitsjningmatrisen med elementene med verdi lik 3. Begge oppfyller de kravene 1,2 og 3 over med hhv  $h = 2$  og  $h = 3$ . Vi har også oppfylt krav 4) fra problemet RTTD, og vi har dermed konstruert en løsning av RTTD.

Vi har bevist at RTTD kan reduseres til TOP-L-k. □

**Korollar 10.** TOP-L-k er NP-komplett.

**Bevis:** Følger av teorem 7 og 8. □

Vi har ikke klart å lage et formelt bevis for det generelle tilfellet TOP-L med  $k < \min(m, n)$ . Vi vet fra 5.1.9 at for  $k = 1$  finner vi optimal løsning i polynomisk tid, så her er problemet åpenbart ikke *NP*-komplett. Vi antar at TOP-L også er *NP*-komplett for de andre verdiene av  $k$ , men et formelt bevis gjenstår som videre arbeid.

## 5.2.2 Polynom løsning av TOP-L-k for uvarierende trafikkmatriser

Vi har vist at SS/TDMA trafikkoptimeringsproblemet for  $k = \min(m, n)$  er *NP*-komplett, men vi skal nå studere en subklasse av trafikkmatriser fra [1] som gir en polynom løsning av TOP-L-k. Begrenset til denne klassen av trafikkmatriser kan vi vise at det finnes en algoritme som konstruerer TDMA skjemaer med antall svisjinger lik den nedre skranken som gitt i korollar 9, og dermed den optimale løsningen.

Vi definerer først klassen av vektete bipartite grafer som kalles *uvarierende* (*unvarying*).

**Definisjon 4.** En vektet bipartitt graf kalles uvarierende hvis antallet kanter av en vilkårlig spesifikk vekt  $w$  intil en vilkårlig node er mindre eller lik antallet kanter av samme vekt som en inntil en node av størst grad.

Trafikkmatrisen som er den reduserte nodeindikatormatrisen til en slik uvarierende graf kaller vi en *uvarierende* matrise.

**Definisjon 5.** En matrise kalles uvarierende dersom antallet elementer av en vilkårlig verdi  $w$  i en vilkårlig rad eller kolonne er mindre eller lik antallet elementer av samme verdi i en rad eller kolonne med flest positive elementer.

Problemet som oppstår for TOP-L-k med uvarierende trafikkmatriser, kaller vi *TOP-L-k-u*. I kapittel 6 skal vi i seksjon 6.2 vise at algoritmen A-TOP-L-u i polynomisk tid gir oss et TDMA skjema med varighet lik den nedre skranken gitt i korollar 9 ([1]). Vi får dermed følgende teorem:

**Teorem 9.** Gitt en  $m \times n$  uvarierende trafikkmatrise  $D$  i et SS/TDMA satellittsystem med  $k = \min(m, n)$  antenner, er den minimale varigheten  $L_0$  av en TDMA frame lik

$$u(D) + v(D)t$$

Algoritmen A-TOP-L-k-u produserer i polynomisk tid en TDMA frame med denne minimale varigheten.

Beviset tas i kapittel 6.2 i forbindelse med algoritmen A-TOP-L-u.

Som et eksempel kan vi se på to trafikkmatriser  $D_1$  og  $D_2$ .  $D_1$  er lik trafikkmatrisen fra eksempelet i kapittel 4.

$$D_1 = \begin{bmatrix} 5 & 0 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}, D_2 = \begin{bmatrix} 3 & 2 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Trafikkmatrisen  $D_1$  fra vårt tidligere eksempel er ikke uvarierende. Antall elementer med verdi 3 i andre rad er større enn i første rad, som er raden med flest positive elementer, og antalle elementer med verdi 2 i tredje rad er større enn i første rad. Vi kan derfor ikke slutte at den nedre skranken  $10 + 3t$  for varigheten av en TDMA frame, som funnet i kapittel 4, kan oppnås.

$D_2$  er derimot uvarierende. Første raden har flest positive elementer, og det er ingen andre rader eller kolonner i matrisen med flere elementer av en spesifikk verdi. En TDMA frame gitt den  $3 \times 4$  trafikkmatrisen  $D_2$  og  $k = \min(m, n) = 3$  har etter korollar 9 en nedre skranke for varigheten  $L$  lik

$$T_0(D) + s_0(D)t = u(D_2) + v(D_2)t = 10 + 4t$$

Ved å benytte algoritmen A-TOP-L-k-u fra seksjon 6.2 får vi følgende TDMA frame, representert ved trafikkmatrisen uttrykt som sum av k-svitsjingsmatriser. Som vi ser er den nedre skranken oppnådd, med  $s = s_0 = 4$  svitsjinger og overføringstid  $T = T_0 = 10$ . Vi har funnet TDMA framen som løser problemet TOP-L-k-u med trafikkmatrisen  $D_2$ .

$$\begin{aligned} D_2 &= \begin{bmatrix} 3 & 2 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

### 5.2.3 Polynom løsning av TOP-L for k-uvarierende trafikkmatriser

I delkapittel 5.2.2 så vi på en klasse trafikkmatriser og korresponderende bipartitte grafer som gav polynome løsninger av SS/TDMA trafikkoptimeringsproblemet TOP-L-k, altså når antall antenner  $k$  er lik det minste

antall sender- eller mottakerstasjoner. Inspirert av det resultatet vil det her presenteres en tilsvarende klasse trafikmatriser og bipartitte grafer som gir polynome løsninger av det generelle problemet TOP-L, der  $k \leq \min n, m$ . Disse matrisene og grafene er en subklasse av de uvarierende matrisene og grafene, og vi kaller dem *k-uvarierende*.

**Definisjon 6.** En vektet bipartitt graf  $G = (X, Y, K)$ , der  $|X| = m$  og  $|Y| = n$ , kalles *k-uvarierende* for en  $k \leq \min(m, n)$  dersom den i tillegg til å være uvarierende også oppfyller følgende krav: Antall kanter av en vekt  $w$  i grafen dividert med  $k$  og avrundet oppover til nærmeste heltall, er mindre eller lik antall kanter av samme vekt  $w$  inntil en node av størst grad.

**Definisjon 7.** En  $m \times n$  matrise  $D$  kalles *k-uvarierende* for en heltallsverdi  $k \leq \min(m, n)$  dersom den i tillegg til å være uvarierende også oppfyller følgende krav: Antall elementer av en vilkårlig verdi  $w$  i matrisen dividert med  $k$  og avrundet oppover til nærmeste heltall, er mindre eller lik antall elementer av samme verdi  $w$  i en rad eller kolonne med flest positive elementer.

Av definisjonen følger at enhver *k-uvarierende* matrise for en gitt  $k$  vil også være *k-uvarierende* for  $k + 1, \dots, \min(m, n)$ . Alle  $m \times n$  uvarierende matriser er dessuten også *k-uvarierende* for  $k = \min(m, n)$ . Det samlede antall elementer av en verdi  $w$  dividert med det minste antallet rader eller kolonner i matrisen, er mindre enn antall elementer av verdien  $w$  i en hvilken som helst rad eller kolonne i matrisen.

Med begrensningen at trafikmatrisen må være *k-uvarierende* gir problem TOP-L opphav til problemet *TOP-L-uk*. Vi skal i kap 6.2 vise at algoritmen A-TOP-L-u fra [1], som gir polynomisk løsning i det uvarierende tilfellet, også for *k-uvarierende* trafikmatriser gir en optimal løsning i polynomisk tid. Vi får dermed følgende teorem:

**Teorem 10.** Gitt en  $m \times n$  *k-uvarierende* trafikmatrise  $D$  i et SS/TDMA satellittsystem med  $k \leq \min(m, n)$  antenner, er den minimale varigheten  $L_0$  av en TDMA frame lik

$$T_0(D) + s_0(D)t = \max\{u(D), \lceil \frac{\sigma(D)}{k} \rceil\} + \max\{v(D), \lceil \frac{\rho(D)}{k} \rceil\}t$$

Algoritmen A-TOP-L-u produserer i polynomisk tid en TDMA frame med denne minimale varigheten.

Beviset gis i kapittel 6.2, og følger av beviset for teorem 9 hentet fra [1].

Som eksempel kan vi se på trafikkmatrisen  $D_2$  fra kapittel 5.2.2 og en ny matrise  $D_3$ . Vi antar et SS/TDMA trafikkoptimeringsproblem med  $k = 2$  antenner.

$$D_2 = \begin{bmatrix} 3 & 2 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, D_3 = \begin{bmatrix} 2 & 2 & 4 & 1 \\ 2 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Trafikkmatrisen  $D_2$  er som tidligere vist uvarierende. Men den er ikke  $k$ -uvarierende for  $k = 2$ . Det er 3 elementer med verdi 1 i  $D_2$ . Dividert med 2 og rundet oppover til 2 er dette større enn antall elementer med verdi 1 i raden med flest positive elementer, rad 1. Vi kan derfor ikke si noe om den nedre skranken for varigheten av en TDMA frame som gitt i korollar 8 kan oppnås.

$D_3$  er også uvarierende. Første rad har flest positive elementer, og det er ingen andre rader eller kolonner i matrisen med flere elementer av en spesifikk verdi.  $D_3$  er i motsetning til  $D_2$  også  $k$ -uvarierende for  $k = 2$ . For alle verdier som finnes i  $D_3$  er det totale antallet elementer av en spesifikk verdi dividert med  $k = 2$  og rundet av opp til nærmeste heltall mindre eller lik antall elementer av samme verdi i første rad. En TDMA frame til en SS/TDMA satellittsystem gitt trafikkmatrisen  $D_3$  og  $k = 2$  har en nedre skranke for varigheten  $L$  lik

$$\begin{aligned} T_0(D_3) + s_0(D_3)t = \\ \max\{u(D_3), \lceil \frac{\sigma(D_3)}{k} \rceil\} + \max\{v(D_3), \lceil \frac{\rho(D_3)}{k} \rceil\}t = \\ \max\{9, \lceil \frac{14}{2} \rceil\} + \max\{4, \lceil \frac{7}{2} \rceil\}t = \\ 9 + 4t \end{aligned}$$

Algoritmen A-TOP-L-u fra kapittel 6.2 vil gi følgende TDMA frame, representert ved trafikkmatrisen som sum av  $k$ -svitjingmatriser. Den nedre skranken for varigheten er oppnådd, med  $s = s_0(D_3) = 4$  svitsjinger og overføringstid  $T = T_0 = 9$ . Denne TDMA frame løser dermed problemet TOP-L-uk med trafikkmatrisen  $D_3$  og antall antenner  $k = 2$ .

$$\begin{aligned} D_3 &= \begin{bmatrix} 2 & 2 & 4 & 1 \\ 2 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Trafikkmatrisen  $D_3$  er dessuten også  $k$ -uvarierende for  $k = 1$ , og vi vil

derfor i polynom tid kunne finne en løsning av TOP-L med  $k = 1$  antenner.

#### 5.2.4 NP-hardhet av tilnærming bedre enn faktor $\frac{7}{6}$ for TOP-L-k

Kompleksitetsresultatene ovenfor gjør at vi vil lete etter gode tilnærmingsalgoritmer for TOP-L og TOP-L-k. Noen slike algoritmer presenteres i kapittel 6. Et viktig spørsmål blir da: Hvor gode kan disse tilnærmingene bli? Med *gap technique* metoden som ble presentert i delkapittel 2.3.2 kan vi vise at det eksisterer nedre skranker for tilnærmingsfaktoren for algoritmene. Det er et NP-hardt problem å tilnærme bedre enn denne nedre skranken.

Fra Crescenzi et al. [7] henter vi beviset for at det er NP-hardt å tilnærme det optimale løsningen av TOP-L-k bedre enn en faktor  $\frac{7}{6}$ . Beviset følger mønsteret til beviset for NP kompletthet av TOP-L-k, med reduksjon fra problemet RTTD, med en noe annen konstruksjon av trafikkmatrisen. Tilsvarende er også vist av Hoogeveen et al. [12] i konteksten av multiprosessor skedulering, men da ved reduksjon fra det NP-komplette problemet *3-satisfiability*.

Vi definerer først en variant av problemet TOP-L-k som skal benyttes i beviset. Gitt et problem TOP-L-k med trafikkmatrise  $D$  og svitsjingt看id  $t$  normaliserer vi elementene i  $D$  med  $t$  og får den nye matrisen  $D'$  (vi må tillate rasjonale verdier i  $D'$ ). Vi definerer da TOP-L-k' til å være SS/TDMA optimeringsproblemet med trafikkmatrise  $D'$  og svitsjingt看id 1. Et TDMA skjema som minimerer varigheten  $L' = T' + s$  i TOP-L-k' (der  $T' = \frac{T}{t}$ ) vil også minimere varigheten  $L = T + st$  i TOP-L-k, og  $L = tL'$ .

**Teorem 11.** *RTTD kan reduseres til problemet TOP-L-k' slik at hvis det finnes en løsning av RTTD er den minmale varigheten av TOP-L-k'  $L' = 6 + 3\epsilon$ , ellers er den optimale løsningen av TOP-L-k lik  $L' \geq 7 + 3\epsilon$ .*

**Bevis:** Gitt en instans  $\langle F, G, A, R \rangle$  av RTTD konstruerer vi en  $2n \times (n+m)$  trafikkmatrise  $D'$  på en liknende måte som i teorem 8.  $\epsilon$  er en vilkårlig liten positiv rasjonalt tall.

1. I den øvre venstre  $m \times n$  submatrisen settes  $d'_{ij} = \epsilon$  dersom  $r_{ij} = 1$  i kravmatrisen  $R$ . Ellers settes elementene lik 0.
2. Fyll den nedre venstre  $m \times n$  submatrisen med 0-elementer.
3. Den høyre  $2n \times n$  submatrisen konstrueres fra tilgjengelighetsfunksjonen  $A$  på følgende måte:

- For  $1 \leq i \leq n$  sett  $d_{ij} = h - 1 + \epsilon$  for alle  $(i, j)$  der  $j = m + 1 + [(i + h - 2) \bmod n]$  og  $A(F_i, h) = 0$  for en  $h \in \{1, 2, 3\}$ .
- For  $n + 1 \leq i \leq 2n$  sett  $d_{ij} = h - 1 + \epsilon$  for alle  $(i, j)$  der  $j = m + 1 + [(i - n + h - 2) \bmod n]$  og  $A(F_{i-n}, h) = 1$  for en  $h \in \{1, 2, 3\}$ .
- Alle andre  $d_{ij}$  settes lik 0.

Som eksempel ser vi på den samme instansen av RTTD som i beviset for teorem 8. Med denne konstruksjonen blir den  $6 \times 7$  trafikkmatrisen  $D'$  seende slik ut:

$$D' = \begin{bmatrix} \epsilon & \epsilon & 0 & 0 & 0 & 1 + \epsilon & 2 + \epsilon \\ 0 & \epsilon & 0 & \epsilon & 0 & \epsilon & 1 + \epsilon \\ \epsilon & 0 & 0 & 0 & 0 & 0 & \epsilon \\ 0 & 0 & 0 & 0 & \epsilon & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 + \epsilon & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 + \epsilon & 2 + \epsilon & 0 \end{bmatrix}$$

Det minimale antalle svitsjinger er som tidligere 3.

Vi antar at vi har en løsning av RTTD. På tilsvarende måte som i det tidligere beviset kan vi vise at elementene i  $D'$  som korresponderer til elementene i kravmatrisen  $R$  for undervisning i time 1, samt elementene i den høyre  $2n \times n$  submatrisen med verdi  $\epsilon$ , utgjør en  $k$ -svitjingmatrise med varighet  $\epsilon$ . Slik finner vi også en  $k$ -svitjingmatrise med varighet  $1 + \epsilon$  og en med varighet  $2 + \epsilon$ . Til sammen utgjør disse en TDMA frame med minimalt antall svitsjinger og overføringstid  $T' = \epsilon + (1 + \epsilon) + (2 + \epsilon) = 3 + 3\epsilon$ . Varigheten av denne TDMA frame blir da lik  $L' = 6 + 3\epsilon$ .

Vi antar så at det ikke finnes noen løsning av vår instans av RTTD, men at det samtidig eksisterer en TDMA frame med minimalt antall svitjinger og overføringstid  $T' < 4 + 3\epsilon$ . For at det skal finnes en slik TDMA frame må alle elementene med verdi  $\epsilon$  fra  $D'$  ligge i samme  $k$ -svitsjingmatrise. Det samme for elementene med verdi  $1 + \epsilon$  og  $2 + \epsilon$ . Dette vil etter samme resonnement som i beviset for teorem 8 bety at vi har en løsning av RTTD. Dermed er  $T' \geq 4 + 3\epsilon$ , og varigheten  $L' \geq 7 + 3\epsilon$ .  $\square$

Ved bruk av *gap technique* fra kapittel 2 får vi da følgende korollar ([7]):

**Korollar 11.** *Det er NP-hardt å tilnærme TOP-L-k med en bedre garantert tilnærmingssratio enn  $\frac{7}{6}$ .*

Dersom det viser seg at TOP-L er NP-komplett også for andre verdier

er  $k$ , vil det i videre arbeid være aktuelt å forsøke å bevise tilsvarende resultat også for disse.

### 5.2.5 Ikke-preemptiv skedulering

Ved ikke-preeemptiv skedulering viser det seg at SS/TDMA trafikkoptimeringsproblemet i en viss forstand er enda hardere å løse enn når vi tillater preemption.

Rendl [19] ser på en generalisert variant av problemet der transponderne kan formidle trafikk mellom flere sende -og mottakerstasjoner i løpet av en svitsjing. I svitsjing  $k$  kan hver sendestasjon sende trafikk til  $k_t$  mottakerstasjoner, og hver mottakerstasjon kan motta trafikk fra  $k_t$  sendere. Det benyttes ikke-preemptiv skedulering. Minimering av overføringstiden  $T$  i dette tilfellet er  $NP$ -komplett. For beviset for dette henvises til [19].

Ved å i tilfellet over la hver  $k_t = 1$ , får vi problemet TOP-T-k-ip. Dermed er også dette problemet  $NP$ -komplett. Vi kan da også slutte at TOP-L-k-ip er  $NP$ -komplett.

Vi skal i kapittel 6 vise at A-TOP-s benytter ikke-preemptiv skedulering for å konstruere et TDMA skjema med minimum antall svitsjinger for TOP-s. En optimal løsning funnet med A-TOP-s er dermed også en optimal løsning av TOP-s-ip.

Som i tilfellet med preemptiv skedulering har vi ikke kunnet gi formelle bevis for at også de generelle tilfellene TOP-T-ip og TOP-L-ip er  $NP$ -komplette.



## Kapittel 6

# Algoritmer

I dette kapitlet presenterer vi algoritmer for løsning av de to delproblemene TOP-s og TOP-T, og heuristikker for tilnærming til en optimal løsning av SS/TDMA trafikkoptimeringsproblemet TOP-L. Vi viser også at algoritmen A-TOP-L-u gir optimal løsning av TOP-L og TOP-L-k gitt henholdsvis  $k$ -uvarierende og uvarierende trafikkmatriser. For TOP-L-k viser vi at det finnes en tilnærmingsalgoritme med en bevisbart god tilnærming.

### 6.1 Algoritmer for TOP-s og TOP-T

Ulikt SS/TDMA trafikkoptimeringsproblemet TOP-L, lar delproblemene TOP-s og TOP-T seg begge løse i polynomisk tid. Vi gir her to algoritmer A-TOP-s og A-TOP-T som konstruerer TDMA skjemaer med hhv. minimum antall svitsjinger og minimum overføringstid, dvs. optimale løsninger av TOP-s og TOP-T. A-TOP-s gir i utgangspunktet kun et skjema med minimalt antall svitsjinger  $s_0$  uten å ta hensyn til overføringstiden  $T$  i skjemaet, mens A-TOP-T gir et skjema med minimum overføringstid  $T_0$  uten å ta hensyn til antall svitsjinger  $s$  det vil medføre.

Algoritmene bygger på samme teknikk, nemlig å finne gjentatte matchinger i den bipartitte grafen som gir trafikkbehovet. Hver matching gir opphav til en  $k$ -svitsjingmatrise. Trafikken i  $k$ -svitsjingmatrisen fjernes fra systemet før det i neste iterasjon på ny lages en matching.

For å sikre minimering av de ønskede variablene gjøres det et par grep der de to algoritmene skiller seg fra hverandre. Først legges det ekstra noder (*dummy nodes*) og ekstra trafikk (*dummy traffic*) til systemet. Deretter gis hver  $k$ -svitsjingmatrise verdier som sikrer at når siste matching er funnet har vi et TDMA skjema med minimum antall svitsjinger eller minimum overføringstid. For bedre å få fram forskjellene mellom

A-TOP-s og A-TOP-T presenteres de ved hjelp av hhv. graf - og matrise-representasjonen av systemet.

I tillegg til å være optimale løsninger av delproblemene TOP-s og TOP-T, kan algoritmene også fungere som heuristikker for TOP-L, med og uten modifikasjoner. A-TOP-s brukes også av algoritmen A-TOP-L-u i delkapittel 6.2. Tilnærmingsalgoritmen A-TOP-L( $\alpha$ ) i delkapittel 6.3.3 gjør bruk av A-TOP-T. De danner således fundamentet for det som presenteres i algoritmekapittelet, og vil derfor forklares inngående.

Fra kapittel 5 har vi fremdeles gjenstående deler av bevisene for teorem 5 og 6, som gav minimumverdier for antall svitsjinger og overføringstid. I bevisene for at A-TOP-s og A-TOP-T terminerer med optimal løsninger gir vi her bevis at de nedre skrankene for  $s$  og  $T$  fra teorem 3 og 4 kan oppnås, og dermed er bevisene for teorem 5 og 6 komplette.

### 6.1.1 A-TOP-s: minimum antall svitsjinger

Hovedideen bak algoritmen A-TOP-s fra [10] er å finne gjentatte maksimum matchinger i den bipartitte grafrepresentasjonen av trafikkmatrisen, der hver matching representerer en  $k$ -svitsjingmatrise i skjemaet. Først legges det til ekstra noder og kanter i grafen, til hver node er av grad  $s_0 = \max\{\nu(D), \lceil \frac{\rho(D)}{k} \rceil\}$ . Ved korollar 1 fra delkapittel 4.3 kan vi kan da garantere at hver maksimum matching er en komplett matching. I hver iterasjon av algoritmen konstrueres det en svitsjing med en kant fra hver node, slik at vi ender opp med et TDMA skjema med  $s_0$  switchinger som gitt i teorem 5.

Algoritmen deler ikke opp kantene i grafen, og benytter seg således av en ikke-preemptiv skeduleringsstrategi. Den løser derfor også TOP-s-ip.

#### Algoritme A-TOP-s

##### Steg 1: Initialisering

Konstruer den bipartitte grafrepresentasjonen  $G_D = (X, Y, K)$  til trafikkmatrisen  $D$ . Regn ut minimum antall svitsjinger  $s_0 = \max\{\nu(D), \lceil \frac{\rho(D)}{k} \rceil\}$ . Sett  $j = 1$ .

##### Steg 2: Addisjon av ekstra noder

Legg noder  $x_{m+1}, \dots, x_{m+n-k}$  til  $X$  og noder  $y_{n+1}, \dots, y_{m+n-k}$  til  $Y$ .

##### Steg 3: Addisjon av ekstra kanter

Legg til kanter med vekt null mellom nodene  $\{x_{m+1}, \dots, x_{m+n-k}\}$  og  $\{y_1, \dots, y_n\}$ , mellom  $\{y_{n+1}, \dots, y_{m+n-k}\}$  og  $\{x_1, \dots, x_m\}$  og mellom  $\{x_1, \dots, x_m\}$  og  $\{y_1, \dots, y_n\}$  til alle noder har grad lik  $s_0$ .

Dette kan gjøres i tre steg:

**Steg 3.1:** For hver node  $x_i \in \{x_{m+1}, \dots, x_{m+n-k}\}$  finn en node  $y_j \in \{y_1, \dots, y_n\}$  med grad mindre enn  $s_0$  og legg til kanten  $(x_i, y_j)$ . Fortsett til alle nodene i  $\{x_{m+1}, \dots, x_{m+n-k}\}$  har grad lik  $s_0$ .

**Steg 3.2:** Tilsvarende for hver node  $y_j \in \{y_{n+1}, \dots, y_{m+n-k}\}$  finn en node  $x_i \in \{x_1, \dots, x_m\}$  med grad mindre enn  $s_0$  og legg til kanten  $(y_j, x_i)$ . Fortsett til alle nodene i  $\{y_{n+1}, \dots, y_{m+n-k}\}$  har grad lik  $s_0$ .

**Steg 3.3:** Til slutt, finn for hver  $x_i \in \{x_1, \dots, x_m\}$  med grad mindre enn  $s_0$ , en node  $y_j \in \{y_1, \dots, y_n\}$  med grad mindre enn  $s_0$ , og legg til  $(x_i, y_j)$ . Fortsett til alle nodene i  $\{x_1, \dots, x_m\}$  og  $\{y_1, \dots, y_n\}$  har grad lik  $s_0$ .

Gi alle de nye kantene vekt null.

**Steg 4:** Konstruer en maksimum matching  $Q$  i  $G_D$

Den vil ha kardinalitet  $m + n - k$ , og er altså en komplett matching i den utvidede grafen.

**Steg 5:** Lag  $k$ -switchingmatrise

Lag  $k$ -switchingmatrisen  $S_j$ , som er den  $m \times n$  reduserte indikatormatrisen til grafen bestående av de  $m + n$  originale nodene og kantene med *ikkenull* vekt i matchingen  $Q$ . Lag ny graf  $G_D$  med kantene  $E = E - Q$ . Hvis  $j < s_0$ , sett  $j = j + 1$  og gå tilbake til Steg 4. Hvis  $j = s_0$ , avslutt.

Algoritmen terminerer med  $s_0$   $k$ -switchingmatriser slik at

$$D = S_1 + S_2 + \dots + S_{s_0}$$

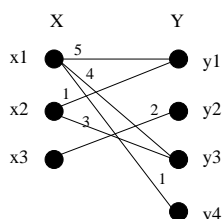
### Eksempel

Som et eksempel på en kjøring av A-TOP-s går vi tilbake til SS/TDMA systemet fra kapittel 4, med  $m = 3$  sendestasjoner,  $n = 4$  mottakerstasjoner og  $k = 2$  antenner. Fra trafikkmatrisen konstruerer vi den bipartitte grafen i figur 6.1 som gir trafikkbehovet. Minimum antall svisjinger er som tidligere  $s_0 = 3$ .

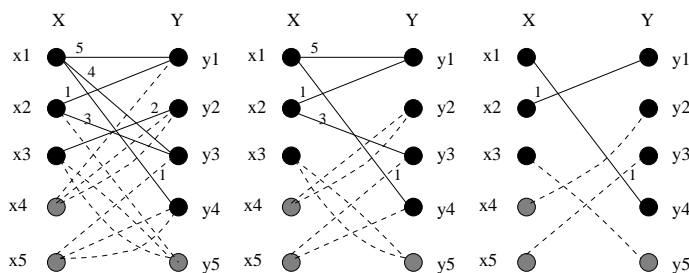
I steg 2 av algoritmen legger vi til ekstra noder  $x_4, x_5$  og  $y_5$  da  $m + n - k =$

5. Deretter legges det i steg 3 til ekstra kanter slik at alle noder har grad lik 3. Vi ender da opp med den første utvidede grafen i figur 6.2. De ekstra nodene er grå, og de ekstra kantene er merket med stiplede linjer.

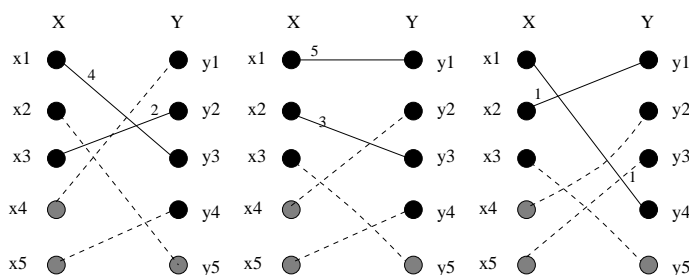
I hver iterasjon av algoritmen finnes det i steg 4 en maskimal matching som i figur 6.3. De originale kantene i disse matchingene gir opphav til svitsjingene i figur 6.4. Trafikken i matchingene trekkes fra de utvidede grafene før det i neste iterasjon på nytt finnes et maksimum matching. Vi ender opp med et TDMA skjema med  $s_0 = 3$  svitsjinger.



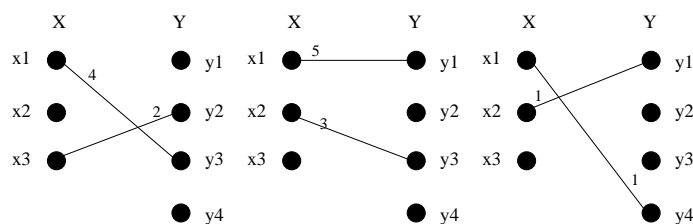
Figur 6.1: Trafikkbehov, bipartitt graf representasjon



Figur 6.2: Utvidede grafer, A-TOP-s iterasjon 1,2,3.



Figur 6.3: Matchinger i de utvidede grafene, A-TOP-s iterasjon 1,2,3.

Figur 6.4: TDMA skjema, svitsjinger  $S_1, S_2$  og  $S_3$ .**Bevis**

Vi må nå bevise at det er mulig å gjøre konstruksjonene i algoritmen ovenfor og at den terminerer med  $s_0$  k-svitsjingmatriser. Beviset er hentet fra [3].

Viser først at det i Steg 3.1 alltid er mulig å finne en node i  $\{y_1, \dots, y_n\}$  som har grad mindre enn  $s_0$ .

Anta det motsatte er sant. Alle nodene  $y_1, \dots, y_n$  har grad lik  $s_0$ . Vi lar  $B$  stå for antall kanter med vekt lik null inntil  $y_1, \dots, y_n$ . Da får vi antall kanter inntil  $y_1, \dots, y_n = ns_0 = \text{antall ikkenull kanter} + \text{antall kanter med vekt lik null inntil } y_1, \dots, y_n$ . Vi lar  $B$  stå for antall kanter med vekt lik null inntil  $y_1, \dots, y_n$ . Da får vi

$$ns_0 = \rho(D) + B$$

Men pga Teorem 1 må vi ha at  $\rho(D) \leq ks_0$ . Dermed får vi

$$B = ns_0 - \rho(D) \geq ns_0 - ks_0 = (n - k)s_0$$

Men siden noen av de  $n - k$  nodene  $x_{m+1}, \dots, x_{n+m-k}$  har grad lavere enn  $s_0$  må vi ha  $B < (n - k)s_0$ , og vi har en selvmodsigelse.

På tilsvarende måte kan vi vise at det i Steg 3.2 alltid er mulig å finne en node i  $\{x_1, \dots, x_m\}$  med grad mindre enn  $s_0$  og at det i Steg 3.3 alltid er mulig å finne en node i  $\{y_1, \dots, y_n\}$  med grad mindre enn  $s_0$ . Konsekvensen er at vi etter Steg 3 har en bipartitt graf der alle nodene har grad  $s_0$ .

Dette muliggjør nå beviset for at vi kan finne en maksimum matching med kardinalitet  $m + n - k$ , altså en komplett matching. Anta det ikke eksisterer en komplett matching. Ved korollar 1 fra delkapittel 4.3 må

det da finnes en delmengde  $A \subset X$  bestående av  $i$  noder som har kanter inntil  $j$  noder i  $Y$  der  $j < i$ . Hvis vi summerer kantene inn til de  $j$  nodene i  $Y$ , får vi at summen blir mindre eller lik  $js_0$ , siden alle nodene i grafen har grad  $s_0$ . Men når vi summerer kantene ut fra  $A$  får vi tilsammen  $is_0$  kanter, som gir at  $is_0 \leq js_0$ . Dette er en selvmotsigelse, så det eksisterer en komplett matching.

Hver iterasjon i Steg 5 fjerner  $m + n - k$  kanter fra grafen, så hver node får graden redusert med 1. I neste iterasjon er altså eksistensen av en matching med kardinalitet  $m + n - k$  fortsatt garantert.

For at matrisen som produseres i Steg 5 skal være en  $k$ -svitsjingsmatrise må den ha høyst  $k$  positive elementer. De andre egenskapene ved en  $k$ -svitsjingsmatrise er ivaretatt ved at matrisen er den reduserte indikatormatrisen til en matching. Siden det gjennom konstruksjonen i Steg 2 ikke er noen kanter mellom  $\{x_{m+1}, \dots, x_{m+n-k}\}$  og  $\{y_{n+1}, \dots, y_{y+n-k}\}$ , må en matching av kardinalitet  $m + n - k$  ha  $(n - k) + (m - k)$  kanter inntil disse nodene. Disse kantene ble alle lagt til i Steg 3 og har derfor vekt null. Da gjenstår det  $m + n - k - (n - k) - (m - k) = k$  kanter som kan være ikkenull.

For å vise at algoritmen terminerer med  $s_0$  kanter, gjenstår da kun å telle hvor mange kanter som fjernes i hver iterasjon. Grafen inneholder  $(m + n - k)s_0$  kanter, og hver matching hvis kanter fjernes fra grafen i Steg 5 er av kardinalitet  $m + n - k$ . Altså ender vi opp med  $s_0$  antall svitsjinger.  $\square$

Vi har nå også bevist det gjenstående fra Teorem 5, den nedre skranken  $s_0$  er et oppnåelig minimum antall svitsjinger for et TDMA skjema.

### Kompleksitet

Tidskompleksiteten av A-TOP-s avhenger av hvilken algoritme som benyttes for å finne maximum matching i steg 4. Den raskeste algoritmen for maximum matching problemet i bipartitte grafer er fra Hopcroft og Karp [13]. Dersom antall noder i grafen er  $O(N)$  og antall antenner  $O(N)$  er denne algoritmen av kompleksitet  $O(N^{\frac{5}{2}})$ . Dersom vi antar det er  $O(N^2)$  kanter i grafen, og det i steg 4 fjernes  $O(N)$  av disse vil det si at, steg 4 gjentas  $O(N)$  ganger. A-TOP-s er dermed av kompleksitet  $O(N^{\frac{7}{2}})$ .

Gopal og Wong oppgir i [10] kompleksitet  $O(n^4)$ , og legger til grunn matchingalgoritmer av  $O(n^3)$  for generelle grafer. Dette tar ikke hensyn

til Hopcroft og Karps resultat for det bipartitte tilfellet.

### 6.1.2 A-TOP-T: minimum overføringstid

Hovedideen bak denne algoritmen fra [4] er svært lik den bak A-TOP-T. Etter å ha addert ekstra trafikk til trafikkmatriksen, og utvidet den med ekstra rader og kolonner, garanterer vi ved korollar 3 at vi kan finne gjentatte *systemer av distinkte representanter* i matrisen, der hvert slikt system representerer en  $k$ -svitsjingsmatrise i TDMA skjemaet. Et system av distinkte representanter får vi ved å finne en maksimum matching i grafrepresentasjonen av trafikkmatriksen.

Verdien av elementene i  $k$ -svitsjingsmatrisene velges slik at summen av overføringstidene blir lik  $T_0 = \max\{u(D), \lceil \frac{\sigma(D)}{k} \rceil\}$ . Resultatet av algoritmen blir da et TDMA skjema med minimal overføringstid  $T_0$ .

#### Algoritme A-TOP-T

##### Steg 1: Initialisering

Gitt trafikkmatrikse  $D$ , regn ut den minimale overføringstiden  $T_0$ . Sett  $R = T_0$  og  $l = 1$ .

##### Steg 2: Addering av ekstra trafikk

Dersom ikke  $D$  er  $k$ -komplett mhp  $R$ , dvs at alle linjesummene er mindre eller lik  $R$  og den totale trafikken er  $\sigma(D) = kR$ , konstruer en ny matrise  $D'$  fra  $D$  på følgende måte:

Sett  $D' = D$ . For hvert element  $d_{ij}$  i  $D'$  legg til

$$q = \min(R - r_i, R - c_j, kR - \sigma(D'))$$

der  $r_i$  og  $c_j$  er hhv radsummen og kolonnesummen knyttet til elementet  $d_{ij}$  i  $D'$ . Mellom hver addisjon oppdateres  $\sigma(D')$ ,  $r_i$  og  $c_j$ . Prosessen terminerer når  $\sigma(D') = kR$ .  $D'$  er nå  $k$ -komplett mhp  $R$ .

##### Steg 3: Utvidelse med ekstra rader og kolonner

Konstruer en  $(m + n - k) \times (m + n - k)$  matrise  $D''$  ved å utvide  $D'$  med  $n - k$  rader og  $m - k$  kolonner med nullelementer.  $D'$  blir da værende submatrisen øverst til venstre i  $D''$ . Den  $(m) \times (m - k)$  submatrisen øverst til høyre kaller vi  $A$ , og den  $(n - k) \times (n)$  submatrisen nederst til venstre kaller vi  $B$ .

For hvert element  $d_{ij}$  i  $A$  og  $B$  legg til

$$q = R - \max(r_i, c_j)$$

og oppdater underveis  $r_i$  og  $c_j$ . Prosessen terminerer nå vi for alle  $d_{ij}$  i  $A$  og  $B$  har enten  $r_i = R$  eller  $c_j = R$ . Da vil også *alle* rad - og kolonne-summer vi  $D''$  være lik  $R$ .

Det skal altså ikke endres andre elementer i  $D''$  enn de i  $A$  og  $B$ . Submatrisen  $D'$  forblir uendret og den  $(n - k) \times (m - k)$  submatrisen nederst til høyre inneholder kun nullelementer.

		<b>n</b>	<b>m-k</b>
<b>m</b>		<b>D'</b>	<b>B</b>
<b>n-k</b>		<b>A</b>	<b>0</b>

Figur 6.5: Utvidet matrise  $D''$  i A-TOP-T

**Steg 4:** Finn et system av distinkte representanter i  $D''$ .

Finn en maksimum matching i den bipartitte grafrepresentasjonen av  $D''$ . Denne vil ha kardinalitet  $m + n - k$ . Elementene i  $D''$  som svarer til kantene i matchingen vil da være et *system av distinkte representanter* i matrisen.  $k$  av elementene vil ligge i submatrisen  $D'$ ,  $\{x_1, x_2, \dots, x_k\}$ .

**Steg 5:** Konstruer  $k$ -svitsjingmatrise

Konstruer den  $m \times n$   $k$ -svitsjingmatrisen  $S_l$  med

$$b_l = \min(R - L, x_1, x_2, \dots, x_k)$$

i plassene til de  $k$  elementene i systemet av distinkte representanter som ligger i submatrisen  $D'$ , der  $L$  er den største linjesummen i  $D'$  som *ikke* er dekket av  $\{x_1, x_2, \dots, x_k\}$ . Resten av elementene i  $S_l$  er null.

Oppdaterer  $D'$  ved å trekke fra elementene i  $k$ -svitsjingmatrisen. Den nye trafikkmatrisen blir da  $D' - S_l$ . Hvis  $D' = [0]$ , gå til Steg 6. Hvis ikke, sett  $l = l + 1$ , oppdater  $R$ , som blir  $R - b_l$ , oppdater den totale trafikken  $\sigma(D')$  i submatrisen  $D'$ , som blir  $\sigma(D') - kb_l = k(R - b_l)$ , og gå tilbake til Steg 3.



**Steg 6:** Trekk fra dummy-trafikk.

Trekk fra  $k$ -svitsjingsmatrisene dummy-trafikken vi la til i Steg 1.

Algoritmen terminerer med et endelig antall  $k$ -svitsjingsmatriser  $S_1, S_2, \dots, S_s$  slik at

$$D = S_1 + S_2 + \dots + S_s$$

Hver  $k$ -svitsjingsmatrise  $S_l$  har maks  $k$  ikkenull elementer med verdi  $b_l$ , og den totale overføringstiden til skjemaet er

$$T = b_1 + b_2 + \dots + b_s = T_0$$

### Eksempel

Vi tar igjen fram eksemplet fra kap 4, med følgende  $3 \times 4$  trafikkmatrise og  $k = 2$  antenner:

$$D = \begin{bmatrix} 5 & 0 & 4 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

I steg 1 av algoritmen beregner vi ut den minimale overføringstiden, og setter  $R$  lik denne. Vi får  $R = 10$ . Deretter settes  $l = 1$ .

Vi setter så  $D' = D$ , og adderer i steg 2 trafikk til  $D$  inntil vi har at summen av elementene  $\sigma(D') = kR = 20$ . Steg 2 resulterer i følgende matrise, der det er addert 4 til element  $d_{2,1}$ .

$$D' = \begin{bmatrix} 5 & 0 & 4 & 1 \\ 5 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

I steg 3 utvider vi  $D'$  med  $n - k = 2$  rader og  $m - k = 1$  kolonne, og adderer ekstra trafikk i de  $(n - k) \times m$  og  $(m - n) \times (n)$  submatrisene som angitt inntil alle linjesummer i den utvidede  $t \times 5$  matrisen  $D''$  er lik  $R$ .

$$D'' = \begin{bmatrix} 5 & 0 & 4 & 1 & 0 \\ 5 & 0 & 3 & 0 & 2 \\ 0 & 2 & 0 & 0 & 8 \\ 0 & 8 & 2 & 0 & 0 \\ 0 & 0 & 1 & 9 & 0 \end{bmatrix}$$

Finner så en maksimum matching i den bipartitte grafrepresentasjonen av  $D''$  i steg 4. Det gir oss følgende system av distinkte representanter:

$$\begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \end{bmatrix}$$

Av disse ligger 2 av elementene i submatrisen  $D'$ ,  $d_{1,1} = 5$  og  $d_{3,2} = 2$ . I steg 5 konstruerer vi derfor k-svitsjingmatrisen  $S_1$  med elementer av verdi 2:

$$S_1 = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

Den nye submatrisen  $D'$  settes lik  $D' - S_1$  og  $R$  settes lik  $R - b = 8$ , før vi går tilbake til steg 4 for å konstruere den nye utvidede matrisen for  $D'$ .

$$D' = \begin{bmatrix} 3 & 0 & 4 & 1 \\ 5 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

I de neste 3 iterasjonene av steg 4 og 5 A-TOP-T konstruerer vi følgende utvidede matriser  $D''$ :

$$\begin{bmatrix} 3 & 0 & 4 & 1 & 0 \\ 5 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 \\ 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 1 & 7 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 4 & 1 & 0 \\ 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Etter fjerning av den ekstra trafikken fra steg 1 ender vi da opp med et TDMA skjema med  $s = 4$  k-svitjingmatriser og overføringstid  $T = T_0 = 10$ , forøvrig den samme som i kapittel 4

$$\begin{aligned} D &= S_1 + S_2 + S_3 + S_4 \\ &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

### Bevis

Skal vise at det er mulig å gjøre konstruksjonene i algoritmen, og at den terminerer med minimal overføringstid  $T_0$ . Beviset er hentet fra [4].

Viser først at når prosessen i Steg 3 terminerer, har faktisk alle rader og kolonner i  $D''$  sum lik  $R$ . Viser først at  $r_i = R$  for  $1 \leq i \leq m$  og  $c_j = R$  for  $n + 1 \leq j \leq m + n - k$ .

Anta det overstående ikke er tilfelle. Da har vi tre mulige tilfeller:

**I:** Anta  $r_i = R \forall i, 1 \leq i \leq m$ , men  $c_j < R$  for en  $j, n + 1 \leq j \leq m + n - k$ . Vi vet at  $\sigma(D') = kR$  og at hver rad i  $D''$  har sum  $R$ . Total trafikk i den  $m \times (m + n - k)$  submatrisen bestående av  $D'$  og  $A$  er da  $mR$ , dvs.  $mR = \sigma(A) + \sigma(D')$  eller

$$\sigma(A) = mR - kR = (m - k)R$$

Siden det er  $m - k$  kolonner i  $A$  kan dermed ingen av dem ha sum mindre enn  $R$ , og vi har en selvmotsigelse.

**II:** Anta  $c_j = R \forall j, n + 1 \leq j \leq m + n - k$ , men  $r_i < R$  for en  $i, 1 \leq i \leq m$ . Igjen følger selvmotsigelsen av at den totale trafikken i de  $m$  første radene i  $D''$  er  $\sigma(A) + \sigma(D') = mR$ . Dette ville ikke vært mulig dersom en av radene hadde sum mindre enn  $R$ .

**III:** Anta det finnes to elementer  $d_{i_1 j_1}$  og  $d_{i_2 j_2}$  slik at  $r_{i_1} = R, c_{j_1} < R$  og  $r_{i_2} < R, c_{j_2} = R$ . Men dette impliserer at elementet  $d_{i_2 j_1}$  har både  $r_{i_2} < R$  og  $c_{j_1} < R$ , som bryter mot antagelsen om at prosessen har terminert.

Ved å bruke samme teknikk på  $B$  kan vi på tilsvarende måte vise at  $c_j = R$  for  $1 \leq j \leq n$  og  $r_i = R$  for  $m + 1 \leq i \leq m + n - k$ . Altså er alle linjesummer i  $D''$  lik  $R$ .  $D''$  er en *kvasi-dobbeltstokastisk matrise*. Vi kan da slutte fra korollar 3 i delkapittel 2.2.2 at  $D''$  har et system av distinkte representanter, bestående av  $m + n - k$  elementer.

Vi har altså et system av  $m + n - k$  distinkte representanter. Da den nedre høyre submatrisen i  $D''$  kun har nullelementer må  $m - k$  representanter være inneholdt i  $A$  og  $n - k$  inneholdt i  $B$ . Det gjenstår da  $(m + n - k) - (m - k) - (n - k) = k$  representanter som er inneholdt i  $D'$ . Matrisen  $S_l$  som konstrueres i Steg 5 er dermed en  $k$ -svitsjingmatrise, siden den er den reduserte indikatormatrisen til en matching og vi nå har vist den har høyst  $k$  positive elementer.

Vi kan også vise at alle linjer i  $D'$  med linjesum  $R$  er dekket av de  $k$  representantene. Vi kaller disse de *kritiske* linjene. Dersom en linje i  $D'$  har sum  $R$  er alle elementene i  $A$  eller  $B$  (etter som om linjen er rad eller kolonne) lik 0. Representanten som ligger på denne linjen er dermed

inneholdt i  $D'$ . Vi vil få bruk for dette når vi skal vise at den nye trafikkmatrisen i neste iterasjon oppfyller de samme kravene som  $D'$  gjorde i denne.

Skal vise at den nye submatrisen  $D'$  etter Steg 5 nå er  $k$ -komplett mhp  $R' = R - b$ , som er den oppdaterte verdien av  $R$ , og at  $R'$  er den minimale overføringstiden til den nye  $D'$ .

Vi skille mellom to mulige tilfeller av  $b_l$ :

**I:** Anta  $b_l = R - L$ . Da har den oppdaterte  $D'$  total trafikk  $\sigma(D') = kR - kb_l = kL$ . Videre viste vi ovenfor at hver kritisk linje i  $D'$  før oppdateringen er dekket av et element fra systemet av distinkte representanter, og har derfor nå sum  $R - b_l = L$ . Linjene i  $D'$  som ikke er dekket av SRD vi fant i Steg 4 forblir uforandret og dermed sum mindre eller lik  $L$ , siden vi definerte  $L$  som den største linjesummen hvis linje ikke er dekket av de distinkte representantene.  $D'$  er dermed  $k$ -komplett mhp  $L = S - b_l$ .

$L$  er også den minimale overføringstiden til den nye  $D'$ . Dette siden  $T_0(D) = \max(r_i, c_j, \lceil \frac{\sigma(D')}{k} \rceil)$  og den maksimale linjesummen nå er  $L$ . Hvis  $T_0$  til den gamle  $D'$  var lik  $\lceil \frac{\sigma(D')}{k} \rceil$ , er nå den totale trafikken redusert med  $kb$  slik at den minimale overføringstiden blir  $\lceil \frac{\sigma(D') - kb_l}{k} \rceil = \lceil \frac{kR - kb_l}{k} \rceil = \lceil R - b_l \rceil = \lceil L \rceil$ .

**II:** Anta  $b_l = x_i$  for en  $i$ . Da har den oppdaterte  $D'$  total trafikk  $\sigma(D') = kR - kx_i = k(R - x_i)$ .  $b_l = x_i$  impliserer at  $R - L \geq x_i$ , siden  $b_l = \min(R - L, x_1, x_2, \dots, x_k)$ . Vi har altså  $R - x_i \geq L$ , og siden  $L$  er den største linjesummen ikke dekket av systemet av distinkte representanter, er alle de ikke-kritiske linjesummene mindre eller lik  $R - x_i$ . Den nye  $D'$  er  $k$ -komplett mhp.  $R - x_i = R - b_l$ . Ved en tilsvarende resonnement som over er  $R - b_l$  også den minimale overføringstiden til  $D'$ .

Siden vi trekker  $b_l$  fra et element i hver linje, er også den oppdaterte  $D''$  en kvasi-dobbelstokastisk matrise, nå med linjesummer  $R - b_l$ . Vi garanteres dermed også i neste iterasjon å finne et system av distinkte representanter i  $D''$ .

Kan nå vise at algoritmen terminerer med minimal overføringstid  $T_0$ .

For hver iterasjon tar vi trafikkmatrise  $D'$  som er  $k$ -komplett mhp  $R$  og med minimal overføringstid  $R$ , konstruerer vi en  $k$ -svitsjingsmatrise  $S_l$  med varighet  $|S_l| = b_l$  og en ny trafikkmatrise  $D' - S_l$ . Den nye tra-

fikkmatrisen er  $k$ -komplett mhp  $R - b_l$ , og har minimal overføringstid  $R - b_l$ . Denne prosessen vil forsette til den endelige trafikkmatrisen har  $k$  ikkenull-elementer igjen. Vi får altså en følge av  $k$ -svitsjingmatriser slik at

$$D = S_1 + S_2 + \dots + S_s$$

Hvis  $R_l$  er den minimale overføringstiden i iterasjon  $l$ , er  $R_{l+1} = R_l - b_l$ . Vi har  $R_1 = T_0$ , og får dermed

$$\begin{aligned} T_0 = R_1 &= b_1 + R_2 = \dots \\ &= b_1 + b_2 + \dots + b_s \end{aligned}$$

□

Dermed er også det gjenstående fra teorem 6 bevist, den nedre skranken  $T_0$  er en oppnåelig minimal overføringstid for et TDMA-skjema.

### Kompleksitet

Anta antall noder i grafen er  $O(N)$  og antall kanter  $O(N^2)$ . Steg 4 kan i verste fall måtte gjentas  $O(N^2)$  ganger. Som nevnt i 6.1.1 er den raskeste algoritmen for maksimum matchinger i bipartitte grafer av kompleksitet  $O(N^{\frac{5}{2}})$ . Algoritmen A-TOP-T er derfor av tidskompleksitet  $O(N^{\frac{9}{2}})$ .

## 6.2 A-TOP-L-u: minimal varighet gitt uvarierende og k-uvarierende trafikkmatriser

I kapittel 5.2.2 presenterte vi de uvarierende trafikkmatrisene som gir polynom løsning av SS/TDMA satelittoptimeringsproblemet TOP-L- $k$  med  $k = \min(n, m)$ . Algoritmen som følger gir et optimalt TDMA skjema, altså med varighet lik den nedre skranken fra korollar 9 ([1]).

Vi definerte i kapittel 5.2.3 de  $k$ -uvarierende trafikkmatrisene, en subklasse av de uvarierende trafikkmatrisene. Disse gir optimal løsning av det generelle problemet TOP-L der  $k \leq \min(n, m)$ . Vi skal vise at algoritmen gir en optimal løsning i polynomisk tid også i dette tilfellet.

### Algoritme A-TOP-L-u

**Steg 1:** La  $|W|$  være antall positive elementer med forskjellige verdier i den uvarierende trafikkmatrisen  $D$ . Del  $D$  i  $a$  matriser  $D_1, D_2, \dots, D_{|W|}$ , der hver matrise  $D_i$  består av elementene fra  $D$  med positiv verdi  $i$ .

**Steg 2:** Finn nøyaktig  $\nu(D_i)$   $k$ -svitsjingmatriser i hver matrise  $D_i$  (med

algoritme A-TOP-s), slik at summen av disse k-svitsjingmatrisene er lik  $D$ . Hver k-svitsjingmatrise svarer til en matching med maksimalt  $k$  kanter i den bipartitte grafen som  $D_i$  er nodeindikatormatrisen til.

**Steg 3:** Trafikkmatrisen  $D$  kan nå skrives som summen av alle k-svitsjingmatrisene funnet i de  $|W|$  matrisene, og representerer en TDMA med optimal løsning av TOP-L-u.

### Bevis for teorem 5.2.2, uvarierende trafikkmatriser

Vi skal her bevise teorem 5.2.2, som sier at algoritmen A-TOP-L-u produserer et TDMA skjema med minimal varighet  $L$ . Beviset er hentet fra [1].

Anta at elementene i  $D$  har verdier  $1, 2, \dots, a$ . Fra steg 1 har vi altså matrisene  $D_1, D_2, \dots, D_a$ , der noen av matrisene kan være tomme.

Anta så at  $r$  er raden eller kolonnen med flest antall elementer. Siden  $D$  er uvarierende, vet vi at det i  $r$  finnes nøyaktig  $\nu(D_1)$  elementer med vekt lik 1,  $\nu(D_2)$  elementer med vekt lik 2, ...,  $\nu(D_a)$  elementer med vekt lik  $a$ .  $\nu(D_i)$  kan være lik 0 for noen  $i$ . Hvis  $u(r)$  er summen av elementene i  $r$  følger det da at  $u(r) = \nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)$ . Dermed får vi

$$u(D) \geq \nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)$$

Det følger også at

$$\nu(D) = \nu(D_1) + \nu(D_2) + \dots + \nu(D_a)$$

Fra korollar 6 har vi at hver matrise  $D_i$  kan deles i  $\nu(D_i)$  k-svitsjingmatriser. Siden alle elementene i  $D_i$  har verdi  $i$ , er summen av overføringstidene til k-svitsjingmatrisene lik  $i\nu(D_k)$ . Varigheten til TDMA skjemaet for hver  $D_k$  er dermed lik  $k\nu(D_k) + \nu(D_k)t$ . Varigheten  $L$  av TDMA skjemaet for  $D$  blir dermed

$$\begin{aligned} & (\nu(D_1) + \nu(D_1)t) + (2\nu(D_2) + \nu(D_2)t) + \dots + (a\nu(D_a) + \nu(D_a)t) = \\ & (\nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)) + (\nu(D_1) + \nu(D_2) + \dots + \nu(D_a))t = \\ & (\nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)) + \nu(D)t \leq \\ & u(D) + \nu(D)t \end{aligned}$$

Fra korollar 9 har vi at  $u(D) + t\nu(D)$  er en nedre skranke for  $L$ . Vi kan dermed slutte at algoritmen er optimal, den produserer et TDMA skjema med minimal varighet  $L = L_0 = u(D) + \nu(D)t$ .  $\square$

### Bevis for teorem 5.2.3, k-uvarierende trafikkmatriser

Teorem 10 i kapittel 5.2.3 sier at algoritme A-TOP-L-u produserer et TDMA skjema med minimal varighet for k-uvarierende trafikkmatriser, og her kommer beviset. Beviset bygger på beviset for teorem 9 i det uvarierende tilfellet, men med de nødvendige modifikasjoner for k-uvarierende trafikkmatriser.

Anta at elementene i  $D$  har verdier  $1, 2, \dots, a$ . Fra steg 1 har vi altså matrisene  $D_1, D_2, \dots, D_a$ , der noen av matrisene kan være tomme.

Siden den k-uvarierende trafikkmatrisen  $D$  implisitt også er uvarierende, følger det som i det uvarierende tilfellet at  $u(D) \geq \nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)$ . Siden  $T_0(D) \geq u(D)$  fra definisjonen av  $T_0(D)$  får vi dermed at

$$T_0(D) \geq \nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)$$

Siden vi også her har at  $\nu(D) = \nu(D_1) + \nu(D_2) + \dots + \nu(D_a)$ , følger det også at

$$s_0(D) \geq \nu(D_1) + \nu(D_2) + \dots + \nu(D_a)$$

Da  $D$  er k-uvarierende har vi for hver  $D_i$  at

$$\lceil \frac{\rho(D_i)}{k} \rceil \leq \nu(D_i)$$

Dette medfører igjen at

$$s_0(D_i) = \max\{\lceil \frac{\rho(D_i)}{k} \rceil, \nu(D_i)\} = \nu(D_i)$$

Fra teorem 5 har vi at hver matrise  $D_i$  kan dekomponeres i  $s_0(D_i) = \max\{\lceil \frac{\rho(D_i)}{k} \rceil, \nu(D_i)\}$  k-svitsjingmatriser. Med k-uvarierende trafikkmatriser finner vi dermed i steg 2  $\nu(D_i)$  k-svitsjingmatriser. Alle elementene i  $D_i$  har verdi  $i$ , og summen av overføringstidene for de  $\nu(D_i)$  k-svitsjingmatrisene blir dermed lik  $i s_0$ . Varigheten til TDMA skjemaet for hver  $D_i$  er dermed lik  $i\nu(D_i) + t\nu(D_i)$ . Den totale varigheten  $L$  av TDMA skjemaet for  $D$  blir dermed

$$\begin{aligned} & (\nu(D_1) + t\nu(D_1)) + (2\nu(D_2) + t\nu(D_2)) + \dots + (a\nu(D_a) + t\nu(D_a)) = \\ & (\nu(D_1) + 2\nu(D_2) + \dots + a\nu(D_a)) + t(\nu(D_1) + \nu(D_2) + \dots + \nu(D_a)) \leq \\ & T_0(D) + t s_0(D) = \max\{\lceil \frac{\sigma(D)}{k} \rceil, u(D)\} + t \max\{\lceil \frac{\rho(D)}{k} \rceil, \nu(D)\} \end{aligned}$$

Dermed har vi vist at vi ender opp med den nedre skranken for  $L$ , altså er algoritmen optimal. A-TOP-L-u produserer et TDMA skjema med minimal varighet  $L = T_0(D) + t s_0(D)$ .  $\square$

### Kompleksitet

Det mest tidskrevende i A-TOP-L-u er steg 2, som benytter algoritme A-TOP-s for å finne  $\nu(D_i)$   $k$ -svitsjingmatriser for hver  $D_i$ . A-TOP-s er av kompleksitet  $O(N^{\frac{7}{2}})$  (se kapittel 6.1.1). Dersom antall noder er av  $O(N)$  og antall kanter  $O(N^2)$  kan steg 2 i verste tilfelle måtte gjentas  $O(N^2)$  ganger. A-TOP-L-u er av tidskompleksitet  $O(N^{\frac{11}{2}})$ .

## 6.3 Heuristikker og tilnærmingsalgoritmer for TOP-L

Vi vil her presentere heuristikker for TOP-L og TOP-L-k. Disse gir intuitivt gode tilnærminger til optimal løsning, men vi kan ikke gi bevis for at de garanterer en viss tilnærmingsratio. For TOP-L-k finnes det også tilnærmingsalgoritmer med bevisbart gode løsninger ([7] og [1]). Vi skal presentere den best av disse, med garantert tilnærmingsratio ekte mindre enn 2 ([1]).

### 6.3.1 A-TOP-s og A-TOP-T som heuristikker

Algoritmene A-TOP-s og A-TOP-T minimerer henholdsvis antall svitsjinger  $s$  og overføringstiden  $T$ , uten å ta hensyn til verdien av det andre elementet i summen som utgjør den totale varigheten  $L$  av skjemaet. Det er likevel interessant å se hvor godt de gjennomsnittlig tilnærmer den minnale varigheten. Vi vil derfor i kapittel 7 foreta en empirisk analyse av algoritmene som heuristikker for TOP-s og TOP-T.

### 6.3.2 A-TOP-s-mod: modifisert A-TOP-s

A-TOP-s har kun som mål å lage et TDMA-skjema med minimum antall svitsjinger. Det kan likevel være interessant å studere et enkelt grep fra [10] som kan redusere overføringstiden  $T$  i skjemaet.

Intuitivt vil vi få kort overføringstid dersom ingen av de  $k$  transponderne ombord i satelitten er uvirksomme i lengre tid i løpet av svitsjtilstandene. Dette bør sikres ved at all trafikk som formidles i løpet av en svitsjtilstand er tilnærmet like stor. I vår matriserepresentasjon vil det si at elementer i trafikkmatrisen av samme størrelsesorden kommer i samme  $k$ -svitsjingmatrise.

Ved anvendelse av en tilføyende vei algoritme for konstruksjonen av en maksimum matching i steg 4, foregår dette, som skissert i 4.3, ved å lage en matching og utvide denne med kanter til vi ikke lenger kan finne en tilføyende vei relativt til matchingen. Ved å ordne kantene



etter avtagende vekt, og kun legge til en kant etter at vi ikke kan finne en tilføyende vei i mengden av kanter med lavere vekt, vil kantene i matchingen fra Steg 4 i hvert fall ha større sprik i størrelsesorden enn uten ordningen. For trafikkmatriser med elementer av svært ulik størrelsesorden bør gevinsten bli merkbart lavere overføringstid i skjemaet, og dermed en bedre tilnærming til den optimale løsningen av TOP-L. Den nye modifiserte heuristikken kaller vi A-TOP-s-mod. Vi skal i den empiriske analysen i kapittel 7 sammenlikne de gjennomsnittlige tilnærmingsratioene for A-TOP-s og

### Eksempel

Vi ser på et eksempel som viser den mulige forskjellen i overføringstid mellom A-TOP-s og A-TOP-s-mod, og hvor god prosentvis forbedring dette kan bety for tilnærmingsratioen  $\frac{L}{L_0}$ .

$$D = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 9 & 1 & 0 \\ 0 & 9 & 5 & 5 & 0 \\ 0 & 5 & 0 & 0 & 5 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Gitt den  $5 \times 5$  trafikkmatrisen  $D$  over, i et system med  $k = 3$  antenner og svitsjingtid  $t = 1$ , vil et TDMA skjema ha nedre skranke  $L_0 = T_0 + s_0 t = 19 + 5 \cdot 1 = 24$ . A-TOP-s produserer følgende TDMA skjema:

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Skjemaet over har varighet  $L = 33 + 5 \cdot 1 = 37$ , og gir dermed tilnærmingsratio  $\frac{L}{L_0} = \frac{37}{24} = 1.54$  i dette tilfellet. Ved bruk av den modifiserte algoritmen A-TOP-s-mod med ordning av kantene, får vi TDMA skjemaet under, med overføringstid  $T = 25$ . Det gir en varighet på  $L = 25 + 5 \cdot 1 = 30$ , og

ny, forbedret tilnærmingsratio  $\frac{L}{L_0} = \frac{30}{24} = 1.25$ . Dette er en forbedring på 18.8%. Som vi ser gir A-TOP-s-mod i forhold til A-TOP-s som forventet et skjema der trafikk av lik størrelse i noe større grad blir formidlet i løpet samme svitsjinger.

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

### 6.3.3 H-TOP-L( $\alpha$ )

Den foreløpig beste tilnærmingsalgoritmen for TOP-L-k presenteres i [1], og har en garantert tilnærmingsratio ekte mindre enn 2. Det gis her en generalisering H-TOP-L( $\alpha$ ) av denne algoritmen, som også heuristisk tilnærmer en optimal løsning av det generelle problemet TOP-L.

Hovedideen bak H-TOP-L( $\alpha$ ) består i å runde av alle elementene i trafikkmatrisen  $D$  oppover til nærmeste multipler av en heltallsverdi  $\alpha$ . Med algoritme A-TOP-T finnes så et TDMA skjema for denne nye trafikkmatrisen  $D_\alpha$ . Vi vil vise at dette TDMA skjemaet med overføringstid  $T_0(D_\alpha)$  kan inneholde maksimalt  $\frac{T_0(D_\alpha)}{\alpha}$  k-svitsjingmatriser, som hver har en overføringstid likt et multiplum av  $\alpha$ .

I den opprinnelige heuristikken fra [1] konstrueres det et TDMA skjema med nøyaktig  $\frac{u(D_\alpha)}{\alpha}$  k-svitsjingmatriser, alle med overføringstid  $\alpha$ . Beviset for at denne konstruksjonen kan gjøres i polynomisk tid bygger på at det bipartitte graffargingsproblemet (*bipartite graph coloring problem*) er løsbart i polynomisk tid. Tilnærmingsalgoritmen fra [1] er dog kun pseudopolynomisk, da kompleksiteten avhenger av  $u(D_\alpha)$  og dermed verdiene av elementene i trafikkmatrisen  $D$ . Det foreslås i artikkelen derfor en modifikasjon tilsvarende algoritme A-TOP-T for  $k = \min(m, n)$ , og dette gjør algoritmen polynomisk. Vi vil bygge beviset for heuristikken H-TOP-L( $\alpha$ ) på konstruksjonene som gjøres i A-TOP-T, og beviset for denne algoritmen.

H-TOP-L( $\alpha$ ) har samme tidskompleksitet som A-TOP-T,  $O(N^{\frac{9}{2}})$ .

**Algoritme H-TOP-L( $\alpha$ )****Steg 1:** Konstruksjon av  $D_\alpha$ 

Gitt trafikkmatrise  $D$ , konstruer ny matrise  $D_\alpha$  ved å runde av verdien for hvert element i  $D$  oppover til et multiplum av en heltallsverdi  $\alpha$ .

**Steg 2:** Finn TDMA skjema for  $D_\alpha$ 

Konstruer et TDMA skjema for  $D_\alpha$  med algoritme A-TOP-T, bestående av maksimalt  $\frac{T_0(D)}{\alpha}$  k-svitsjingmatriser. Dette vil også være et TDMA skjema for trafikkbehovet gitt ved  $D$ .

**Eksempel**

Vi skal se på et eksempel med en  $4 \times 4$  trafikkmatrise  $D$ ,  $k = 3$  antenner og svitsjingtid  $t = 2$ .

$$D = \begin{bmatrix} 5 & 3 & 6 & 7 \\ 0 & 6 & 8 & 3 \\ 3 & 0 & 0 & 6 \\ 4 & 2 & 0 & 0 \end{bmatrix}, D_2 = \begin{bmatrix} 6 & 4 & 6 & 8 \\ 0 & 6 & 8 & 4 \\ 4 & 0 & 0 & 6 \\ 4 & 2 & 0 & 0 \end{bmatrix}$$

Med  $\alpha = 2$  gir steg 1 i H-TOP-L(2) matrisen  $D_2$ , der alle elementene har verdi lik multipler av 2. Heuristikken produserer følgende TDMA skjema med  $s = 6$  svitsjinger og overføringstid  $T = 24$ :

$$\begin{aligned} D_2 &= S_1 + S_2 + S_3 + S_4 + S_5 + S_6 \\ &= \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 4 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Varigheten av skjemaet blir da  $L = T + st = 24 + 6 \cdot 2 = 36$ . Den nedre skranken for varigheten er  $L_0(D) = T_0(D) + s_0(D)t = 21 + 4 \cdot 2 = 29$ , som gir en tilnærmingsratio

$$\frac{L}{L_0(D)} = \frac{36}{29} = 1.24$$

### Bevis

Vi skal her vise at H-TOP-L( $\alpha$ ) gir et TDMA skjema med maksimalt  $\frac{T_0(D_\alpha)}{\alpha}$  svitsjinger. Dette resultatet er grunnlaget for den gode tilnærmingsratioen vi skal bevise senere.

Først viser vi at hver av k-svitsjingmatrisene vil ha overføringstid likt et multiplum av  $\alpha$ . I steg 5 av algoritme A-TOP-T konstrueres det en k-svitsjingmatrise med elementer av verdi  $b = \min\{R - L, x_1, x_2, \dots, x_k\}$ . Alle  $x_i$  er elementer i  $D'_\alpha$ , og dermed multipler av  $\alpha$  pluss evt. ekstra trafikk fra steg 2.  $L$  er en linjesum i  $D'_\alpha$ , og dermed også et multiplum av  $\alpha$  pluss evt. ekstra trafikk.  $R$  er i første iterasjon lik  $T_0(D_{alpha})$ , siden trekkes det i hver iterasjon fra  $b$ . Fra konstruksjonen av  $D_\alpha$  fra  $D$  følger det at  $T_0(D_\alpha) = \max\{u(D_\alpha), \lceil \frac{\sigma(D_\alpha)}{k} \rceil\}$  også er et multiplum av  $\alpha$ , da både den maksimale linjesummen  $u(D_\alpha)$  og den totale summen  $\sigma(D_\alpha)$  av elementene i  $D_\alpha$  er multipler av  $\alpha$ . Altså er  $b$  likt et multiplum av  $\alpha$  pluss ekstra trafikk fra steg 1 i hver iterasjon av A-TOP-T. Nå vi i steg 6 av A-TOP-T trekker fra den ekstra trafikken ender vi følgelig opp med k-svitsjingmatriser som alle har overføringstider lik multipler av  $\alpha$ .

En øvre skranke for antall svitsjinger i et TDMA skjema produsert med H-TOP-L( $\alpha$ ), finner vi i tilfellet der hver av k-svitsjingmatrisene har overføringsstid lik  $\alpha$ . Siden A-TOP-T benyttet på  $D_\alpha$  gir en overføringstid lik  $T_0(D_\alpha)$ , vil et slikt skjema bestå av  $\frac{T_0(D_\alpha)}{\alpha}$  k-svitsjingmatriser.  $\square$

### Tilnærmingsratio

For tilfellet der  $k = \min(m, n)$  har vi et resultat fra [1], som sier at for riktig valg av  $\alpha$  gir H-TOP-L( $\alpha$ ) en tilnærmingsratio ekte bedre enn 2 for varigheten  $L$  av et TDMA skjema. Dette er den foreløpig beste tilnærmingsratioen for problemet TOP-L-k.

**Teorem 12.** *H-TOP-L( $t+1$ ) gir en tilnærming av problemet TOP-L-k innen  $2 - \frac{1}{t+1}$  av den nedre skranken.*

**Bevis:** Med overføringstid  $u(D_\alpha)$  og maksimalt  $\frac{u(D_\alpha)}{\alpha}$  svitsjinger får et TDMA skjema for TOP-L-k laget med H-TOP-L( $\alpha$ ) en total varighet

$$L \leq u(D_\alpha) + \frac{u(D_\alpha)}{\alpha} t.$$

Vi kan dessuten vise at  $u(D_\alpha) \leq u(D) + (\alpha - 1)v(D)$ . Anta at det motsatte var tilfellet. En av linjesummene vil da være økt med mer enn  $(\alpha - 1)v(D)$ . Dermed må også minst ett av elementene i  $D$  være økt med mer enn  $(\alpha - 1)$  i konstruksjonen av  $D_\alpha$ . Vi antar dette elementet i  $D$  har

verdi lik  $a$ . I  $D_\alpha$  har samme element verdi lik et helt multiplum av  $\alpha$ ,  $b\alpha$ . Vi påstår at  $b\alpha - a > \alpha - 1$ . Dette er ekvivalent med at  $a < (b - 1)\alpha + 1$ . Siden  $a$  er heltall vil dette si at  $a \leq (b - 1)\alpha$ . Dette er en selvmotsigelse, da  $a$  i dette tilfellet ville vært rundet opp til  $(b - 1)\alpha$ , og ikke  $b\alpha$ .

Det følger da at tilnærmingsratioen den varigheten  $L$  er begrenset oppad på følgende vis:

$$\begin{aligned} L &\leq u(D_\alpha) + \frac{u(D_\alpha)}{\alpha}t \\ &\leq u(D) + (\alpha - 1)v(D) + \frac{u(D) + (\alpha - 1)v(D)}{\alpha}t \\ &= \frac{t + \alpha}{\alpha}(u(D) + (\alpha - 1)v(D)) \end{aligned}$$

Ved å la  $\alpha = t + 1$  får vi da at at H-TOP-L( $t + 1$ ) gir et TDMA skjema med tilnærmingsratio

$$\frac{L}{L_0} \leq \frac{2t + 1}{t + 1} = 2 - \frac{1}{t + 1}$$

□

Dessuten har vi en preposisjon fra [1] som sier at H-TOP-L( $\alpha$ ) gir den beste tilnærmingsratioen for  $\alpha = t + 1$ . For beviset henvises til artikkelen.

**Preposisjon 6.1.** *Det finnes ingen  $\alpha$  som med H-TOP-L( $\alpha$ ) gir bedre tilnærmingsratio enn  $2 - \frac{1}{t+1}$  for problemet TOP-L- $k$ .*

Men hva med det generelle tilfellet  $k \leq \min(m, n)$ ? Vi kan ikke vise en tilsvarende øvre skranke for tilnærmingsratioen i TOP-L. Vi vil derfor i kapittel 7 vurdere hvordan H-TOP-L( $\alpha$ ) med forskjellige verdier av  $\alpha$  fungerer som heuristikk for TOP-L for ulike verdier av  $k$  og  $t$ .



## Kapittel 7

# Beregningsresultater

I dette kapittelet presenterer vi flere beregningsresultater av arbeidet gjort i oppgaven. For et utvalg tilfeldig genererte matriser undersøker vi den prosentvise forekomsten av uvarierende og k-uvarierende matriser. Vi presenterer så en empirisk analyse av tilnærmingsratioen for heuristikkene i oppgaven.

### 7.1 Implementasjon

I implementasjonen av heuristikkene presentert i oppgaven er det lagt vekt at koden på en oversiktlig måte gjenspeiler heuristikkenes hovedideer. Det er ikke lagt vekt på hastighet og optimalisering av kode. Dette gjenspeiles i både valg av verktøy og algoritme for å finne maksimum matching (steg 4 i A-TOP-s og A-TOP-T).

Heuristikkene er implementert i det numeriske beregningssystemet MATLAB fra Mathworks. MATLAB er et kraftig system for behandling av vektorer og matriser, med et høynivå *funksjonelt* programmeringsspråk ([21]). Funksjonene og operatorene i språket er dessuten *vektoriserte*, dvs. at de inneholder løkker for å iterere over elementene i vektor inputparametere. I egen kode som benytter MATLABs standardbibliotek, skal det således være mindre behov for å bruke løkker over vektor - og matri-seelementer. På denne måten kommer ideene i heuristikkene tydeligere fram enn ved implementasjon i et laverenivå programmeringsspråk.

Sentralt i alle heuristikkene står problemet med å finne maksimum matching i en bipartitt graf. Implementasjonen av algoritmer for dette (som presentert i kapittel 4.3) ligger ikke innenfor hovedfokuset til denne oppgaven, og det er derfor valgt å benytte en funksjon fra biblioteket i MATLAB. Funksjonen *dmperm* gir en *Dulmage-Mendelsohn dekomposisjon* av matrisen, og benyttes til å permutere matrisen til en *block upper*

*triangular form* som bl.a. letter løsningen av visse lineære likningssystemer ([18]). En del av denne dekomposisjonen er å finne en maksimum matching i den bipartitte grafen som matrisen er den reduserte indikatormatrisen til. Det framgår ikke av dokumentasjonen til MATLAB biblioteket hvilken algoritme som benyttes for å finne maksimum matching, og vi velger derfor å anvende funksjonen som en *black box*<sup>1</sup> der vi kun er interesserte i resultatet. Hvis  $G$  er en bipartitt graf og  $D$  den reduserte indikatormatrisen til  $G$ , gir  $p = dmperm(D)$  posisjonene til elementer i  $D$  som tilsvarer kantene i en maksimum matching i  $G$ .

Ved å benytte *dmperm*-funksjonen for å finne maksimum matching tvinges vi også til å benytte en annen strategi enn beskrevet i 6.3.2 for å implementere modifikasjonen av A-TOP-s som gir lavere overføringstid i algoritme A-TOP-s-mod. Vi kan ikke velge ordningen av kantene matching-algoritmen leter etter tilføyende vei i. A-TOP-s-mod er derfor implementert på en alternativ måte i denne oppgaven. I steg 4 av A-TOP-s-mod konstrueres en ny graf med kun de nye kantene lagt til i steg 3. De opprinnelige kantene ordnes så etter vekt, og kantene med størst vekt legges til den nye grafen. Med *dmperm* finner vi så en maksimum matching i grafen. Da alle nodene nå ikke har lik grad kan vi ikke garantere at matchingen er komplett. Dersom matchingen ikke er komplett legger vi til de gjenstående opprinnelige kantene av størst vekt, og finner igjen en maksimum matching med *dmperm*. Dersom vi ikke finner komplette matchinger kan vi gjenta dette inntil vi har lagt til alle de opprinnelige kantene. Da finner vi garantert en komplett matching, og kan gå videre til steg 5 for å konstruere en k-svitsjingmatrise. Dersom vi finner en komplett matching før alle de opprinnelige kantene er lagt til, har vi garantert en matching med kanter av forhåpentligvis noe likere vekt enn matchingen vi finner for grafen med alle de opprinnelige kantene. Resultatet blir som i 6.3.2 en heuristikk som finner minimum antall svitsjinger, og i beste fall, lavere overføringstid enn A-TOP-s.

Valgene ovenfor har følger for effektiviteten av de implementerte algoritmene. Tidsbruken vil sannsynligvis kraftig reduseres ved anvendelse av et programmeringsspråk på et noe lavere nivå og en dedikert egenimplementert matchingalgoritme. Vi har derfor ikke lagt stor vekt på kjøretider i beregningsresultatene. Fokuset i den empiriske analysen vil ligge på kvaliteten av heuristikkene for TOP-L, altså hvor godt de tilnærmer en optimal løsning av SS/TDMA trafikkoptimeringsproblemet.

Programkoden til implementasjonen finnes i tillegg A.

---

<sup>1</sup>Funksjon der vi kun kjenner data inn og data ut, og ikke hvordan utdata ble beregnet



## 7.2 Forekomster av uvarierende og k-uvarierende trafikkmatriser

Vi har presentert klasser av trafikkmatriser som muliggjør løsning av TOP-L i polynom tid. Men hvor vanlige er de uvarierende og k-uvarierende trafikkmatrisene? For å kunne si noe om hvorvidt vi i praksis vil kunne ha nytte av de teoretiske resultatene, må vi vite noe om de prosentvise forekomstene av slike trafikkmatriser i SS/TDMA trafikkoptimeringsproblemer.

I mangel på tall fra reelle satellittsystemer kan man simulere statistikken ved å se på tilfeldig genererte trafikkmatriser, se tabell 7.1. Hver  $m \times n$  trafikkmatriks har som elementer heltall med verdi mellom 0 og  $d_{max}$ . Verdiene er tilfeldig generert med uniform distribusjon. For hvert sett av parametere er det generert 20000 matriser og sjekket hvor mange av disse som er uvarierende eller k-uvarierende for  $k = 1, \dots, \min(m, n)$ . Målingene er også gjort for tilfeldige genererte *sparse*<sup>2</sup> matriser med tetthet 0.5, også her med uniformt fordelte verdier mellom 0 og  $d_{max}$ . Programkoden er vedlagt i tillegg A.

Som vi ser av tallene i figur 7.1 er det prosentvis svært få av de tilfeldig genererte matrisene som er uvarierende eller k-uvarierende. Fra simuleringene kan det se ut som det er tre faktorer som i hovedsak bestemmer hyppigheten av uvarierende og k-uvarierende matriser: dimensjon, antall ulike verdier og antall nullelementer. Ettersom dimensjonene og antallet mulige ulike verdier i matrisene øker, avtar antallet uvarierende og k-uvarierende matriser. Sparse matriser med mange nullelementer gir flere tilfeller av (k-)uvarierende matriser med ellers lik dimensjon og verdispenn.

Siden de tilfeldig genererte trafikkmatrisene ikke nødvendigvis gjenspeiler reelle trafikkbehov i SS/TDMA satellittsystemer, er det vanskelig å trekke entydige konklusjoner fra dette tallmaterialet. Det ser dog ut som at det er en dårlig strategi å satse på at en betydelig andel SS/TDMA trafikkoptimeringsproblemer vil kunne løses optimalt, på bakgrunn av at trafikkbehovet gir uvarierende eller k-uvarierende trafikkmatriser.

## 7.3 Empirisk analyse av heuristikkene

Når vi skal evaluere heuristikkene for TOP-L er vi interessert i hvor god tilnærmingen er for de gjennomsnittlige trafikkbehovene som oppstår

---

<sup>2</sup>Matrise med prosentvis få antall ikkenull-elementer

$m = 3, n = 4, d_{max} = 4$				
			Sparse	
	Antall	Prosent	Antall	Prosent
Ikke-uvarierende	19840	99.20%	15400	77.00%
1-uvarierende	0	0	689	3.49%
2-uvarierende	22	0.11%	4162	20.81%
uvarierende	160	0.80%	4600	23.00%

$m = 3, n = 4, d_{max} = 30$				
			Sparse	
	Antall	Prosent	Antall	Prosent
Ikke-uvarierende	20000	100%	19639	98.20%
1-uvarierende	0	0	150	0.75%
2-uvarierende	0	0	351	1.76%
uvarierende	0	0	361	1.80%

$m = 10, n = 10, d_{max} = 4$				
			Sparse	
	Antall	Prosent	Antall	Prosent
Ikke-uvarierende	20000	100%	19994	99.970%
4-uvarierende	0	0	1	0.005%
5-uvarierende	0	0	1	0.005%
6-uvarierende	0	0	3	0.015%
7-uvarierende	0	0	5	0.025%
8-uvarierende	0	0	6	0.030%
9-uvarierende	0	0	6	0.030%
uvarierende	0	0	6	0.030%

$m = 10, n = 10, d_{max} = 30$				
			Sparse	
	Antall	Prosent	Antall	Prosent
Ikke-uvarierende	20000	100%	20000	100%

Tabell 7.1: Forekomster av uvarierende og k-uvarierende matriser blant 20000 tilfeldig genererte matriser.

i et SS/TDMA satellittsystem. I denne empiriske analysen beregnes den gjennomsnittlige ytelsen til algoritmen, og dette forteller hvordan algoritmen oppfører seg “normalt”. Ytelsen måles i tilnæringsratioen  $\frac{L}{L_0}$ .

I mangel på data fra reelle SS/TDMA satellittsystemer, simulerer vi trafikkbehov ved å generere tilfeldig valgte trafikkmatriser og kjøre heuristikene på disse. Det genereres 200 tilfeldige trafikkmatriser. Hver matrise er av dimensjon  $10 \times 10$  med tilfeldig valgte elementer  $0 \leq d_{ij} \leq 30$ . For hver matrise kjøres alle heuristikene for  $k = 1, 2, 3, 4, 5, 6, 7, 8, 10$ . H-TOP-L( $\alpha$ ) kjøres i tillegg for  $1 \leq \alpha \leq 30$ . For hver av algoritmene beregnes så tilnæringsratioen  $\frac{L}{L_0}$  for svitsjingtider  $1 \leq t \leq 100$ . For hvert sett  $\langle k, t \rangle$  beregnes deretter gjennomsnittlig tilnæringsratio over de 200 trafikkmatrisene for hver algoritme (for H-TOP-L( $\alpha$ ) for hvert sett  $\langle k, t, \alpha \rangle$ ).

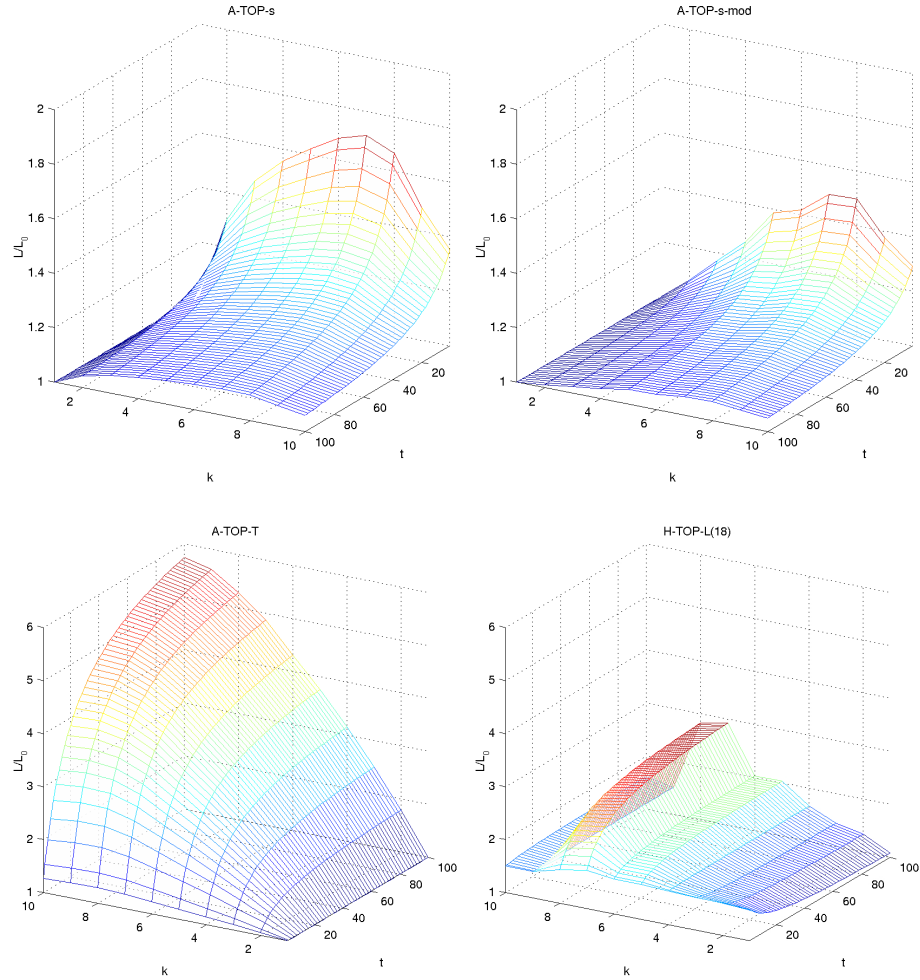
Vi skal først se litt på noen plott av de gjennomsnittlige tilnæringsratioene for de ulike algoritmene. Deretter skal vi sammenlikne A-TOP-s og A-TOP-s-mod for å se effekten av de ordnede kantene. For H-TOP-L( $\alpha$ ) skal vi vurdere valg av  $\alpha$  når  $k < \min(m, n)$ . Vi skal så forsøke å velge den “beste” heuristikken ut fra kriteriet om best gjennomsnittlig tilnæringsratio. Til slutt skal vi kort se på kjøretider for algoritmene.

### 7.3.1 Gjennomsnittlig tilnæringsratio, alle algoritmer

Tilnæringsratio $\frac{L}{L_0}$						
Algoritme	t	k				
		2	4	6	8	10
A-TOP-s	1	1.30	1.55	1.68	1.67	1.35
	100	1.04	1.08	1.09	1.10	1.06
A-TOP-s-mod	1	1.08	1.26	1.41	1.50	1.29
	100	1.01	1.04	1.06	1.07	1.05
A-TOP-T	1	1.05	1.15	1.24	1.30	1.27
	100	1.79	3.16	4.31	5.28	5.74
H-TOP-L(18)	1	1.61	1.64	1.66	1.64	1.51
	100	1.34	1.66	2.03	2.83	1.62

Tabell 7.2: Gjennomsnittlige tilnæringsratioer for  $t = 1$  og  $t = 100$ .

Vi har kjørt alle tilnæringsalgoritmene for de tilfeldige genererte trafikkmatrisen som beskrevet over. Figur 7.1 illustrerer hvordan den gjennomsnittlige tilnæringsratioen varierer som funksjon av antall anten-



Figur 7.1: Gjennomsnittlig tilnærmingsratio  $\frac{L}{L_0}$  som funksjon av  $k$  og  $t$  for algoritmene A-TOP-s, A-TOP-s-mod, A-TOP-T og H-TOP-L(18)

ner  $k$  og svitsjingtiden  $t$ . For H-TOP-L( $\alpha$ ) har vi valgt å plote for  $\alpha = 18$ . I tabell 7.2 fås tall på tilnærmingsratioene for  $t = 1$  og  $t = 100$  for alle verdier av  $k$ .

Det er i plottene i figur 7.1 gjort endel grep for å bedre få fram detaljene. For det første er tilnærmingsratioen plottet som et tredimensjonalt *mesh* som funksjon av  $t$  og  $k$ . Dette gir et noe uriktig bilde av  $\frac{L}{L_0}$  som en kontinuerlig funksjon av både  $t$  og  $k$ . Det er kun som funksjon av  $t$  gitt en  $k \in \{1, 2, 3, 4, 5, 6, 7, 8, 19\}$  tilnærmingsratioen er kontinuerlig. For en gitt svitsjingtid  $t$  er  $\frac{L}{L_0}$  en diskret funksjon av  $k$ . Dernest er det valgt forskjellig størrelse på ratio-aksene i plottene for A-TOP-s/A-TOP-s-mod

og A-TOP-T/H-TOP-L(18). Vi ser også plottene fra forskjellige vinkler, og aksene for  $k$  og  $t$  går derfor i forskjellige retninger.

Vi vil først kommentere tilnærmingsratioen som funksjon av svitsjingtiden  $t$  gitt en fast  $k$ . Her ser vi bort fra tilfellet  $k = 1$ , da vi for denne verdien av  $k$  oppnår den nedre skranken for varigheten for alle verdier av  $t$ , og tilnærmingsratioen blir således lik 1. Vi ser klart av figur 7.1 at algoritmene A-TOP-s og A-TOP-s-mod oppfører seg anderledes enn A-TOP-T og H-TOP-T(18). A-TOP-s og A-TOP-s-mod har begge høyest tilnærmingsratio for  $t = 1$ , og ratioen avtar når svitsjingtiden øker. Varigheten  $L = T + st$  kan jo ses på som en vektet sum av  $T$  og  $s$ . Med økende verdier av  $t$  øker vi betydningen av  $s$  i den vektete summen, og A-TOP-s og A-TOP-s-mod gir bedre tilnærminger til den nedre skranken  $L_0 = T_0 + s_0t$ .

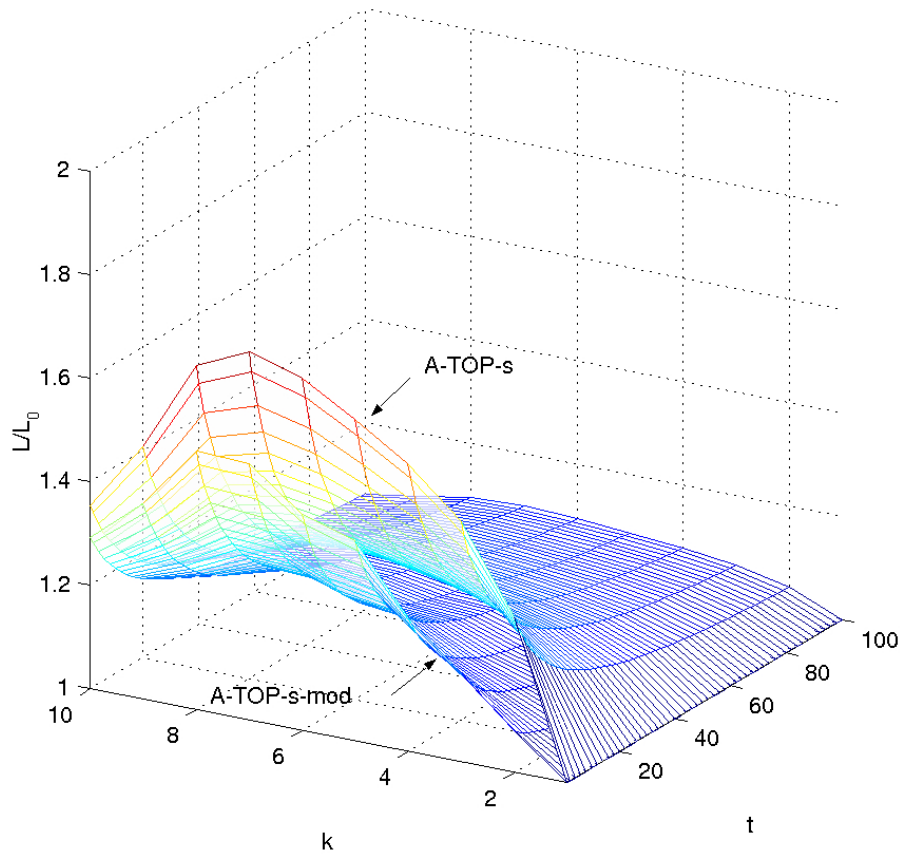
Algoritmen A-TOP-T oppfører seg på direkte motsatt måte. Beste tilnærmingsratio finnes for  $t = 1$ . Med økende  $t$  øker også  $\frac{L}{L_0}$ . Siden denne algoritmen minimerer overføringstiden  $T$  til TDMA skjemaet og ikke tar hensyn til antall svitsjinger  $s$ , vil den økte betydningen av  $s$  i den vektete summen medføre at A-TOP-T gir dårligere tilnærminger til den nedre skranken.

H-TOP-L( $\alpha$ ) benytter algoritme A-TOP-T, og følgelig viser H-TOP-L(18) flere av de karakteristiske egenskapene til denne algoritmen. Men den har likevel en betraktelig mer kompleks oppførsel. For noen verdier av  $k$  øker tilnærmingsratioen med  $t$  på samme måte som A-TOP-T, mens den for andre ikke lar seg analysere med samme letthet. Den komplekse oppførselen som oppstår som følge av introduksjonen av den tredje parameteren  $\alpha$  skal vi se litt nærmere på i 7.3.3, når vi vurderer hvilke valg av  $\alpha$  som gir best tilnærmingsratio for H-TOP-L( $\alpha$ ) gitt ulike verdier av  $k$  og  $t$ .

### 7.3.2 Sammenlikning av A-TOP-s og A-TOP-s-mod

For å avgjøre hvilke følger modifikasjonen av A-TOP-s i 6.3.2 har for egnetheten av algoritmen som heuristikk for problemet TOP-L, skal vi sammenlikne gjennomsnittlig og dårligste tilnærmingsratio for A-TOP-s og A-TOP-s-mod anvendt på de tilfeldig genererte trafikkmatrisene.

Figurene 7.2 og 7.4 viser plott av gjennomsnittlige tilnærmingsratio som funksjon av  $t$  og  $k$ . Som vi ser gir A-TOP-s-mod gjennomgående bedre ratio enn A-TOP-s for alle verdier av  $t$  og  $k$  (untatt  $k = 1$ ). Dersom vi unnlater tallmaterialet for  $k = 1$ , varierer for ulike verdier av  $k$  og  $t$  den



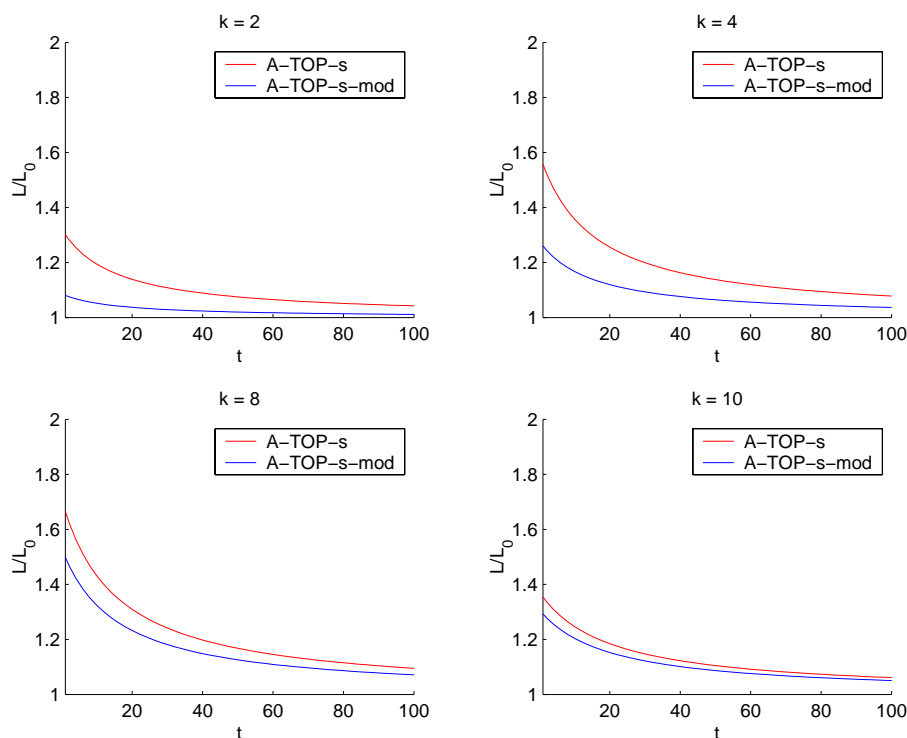
Figur 7.2: A-TOP-s og A-TOP-s-mod, gjennomsnittlig tilnæringsratio  $\frac{L}{L_0}$  som funksjon av  $t$  og  $k$ .

prosentvise forbedringen fra 1% til 30% av nedre skranke  $L_0$ . Som vi ser av figuren avtar forbedringen når  $k$  vokser. Med økende  $k$  avtar antall svitsjinger, og effekten av å overføre trafikk av samme størrelse i samme svitsjinger avtar tilsvarende.

Vi konkluderer av dette at bruken av A-TOP-s-mod som heuristikk for TOP-L innebærer en merkbar forbedring i forhold til A-TOP-s.

### 7.3.3 Valg av $\alpha$ for H-TOP-L( $\alpha$ )

I kapittel 6.3.3 viste vi at H-TOP-L( $\alpha$ ) gav en tilnærming garantert ikke mindre enn en faktor 2 av den nedre skranken for A-TOP-L- $k$  med  $k = \min(m, n)$ . Den beste garanterte tilnærmingen fikk vi for  $\alpha = t + 1$ . Når vi så ønsker å analysere ytelsen til H-TOP-L( $\alpha$ ) empirisk, antar vi

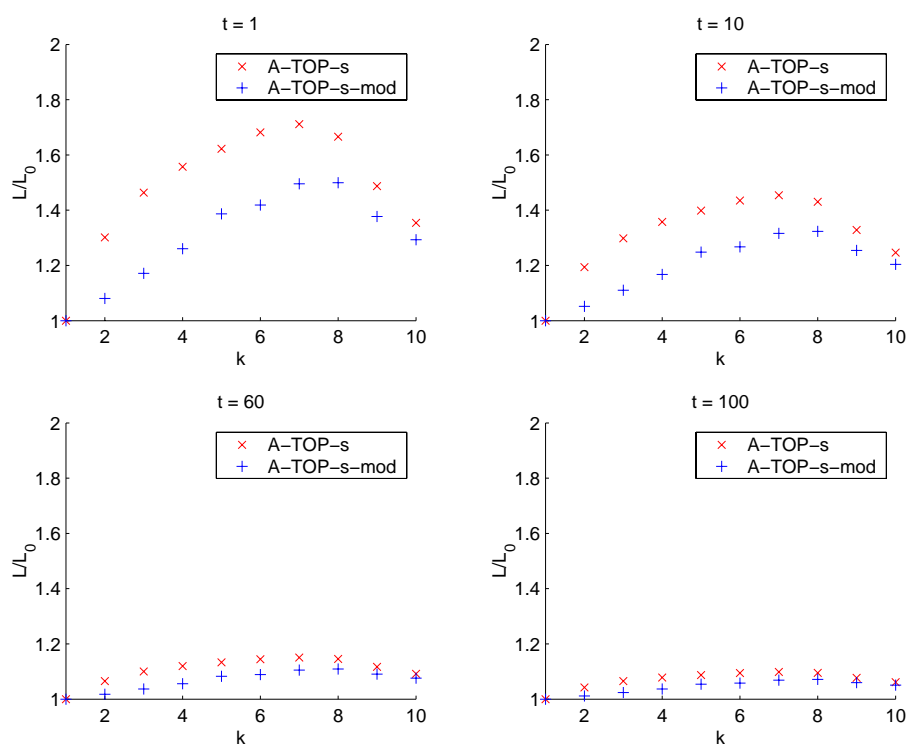


Figur 7.3: A-TOP-s og A-TOP-s-mod, gjennomsnittlig tilnæringsratio  $\frac{L}{L_0}$  som funksjon av svitsjingtid  $t$ .

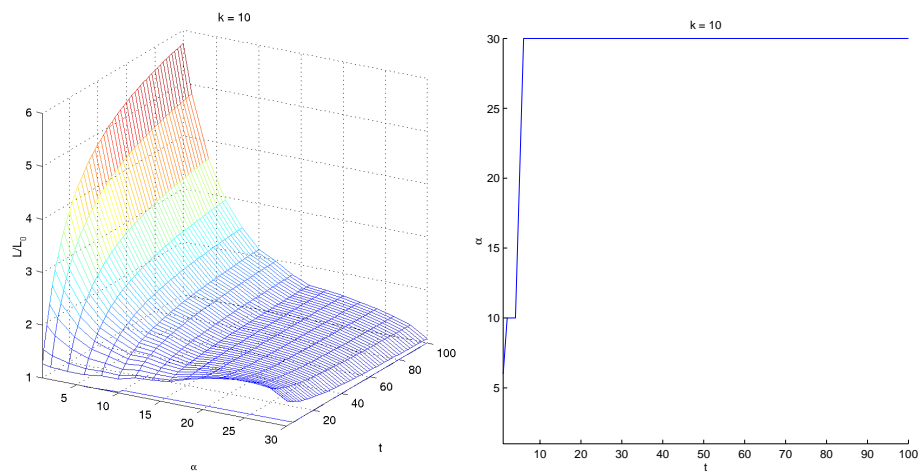
dermed at vi også for å finne den beste gjennomsnittlige tilnæringsratioen må velge riktig  $\alpha$  i forhold til svitsjingtiden  $t$ . Vi ønsker også å se på tilnærming av det mer generelle problemet TOP-L, og vi vil derfor undersøke om beste valg av  $\alpha$  er avhengig av antallet antenner  $k$ .

I figurene 7.5, 7.6 og 7.7 er den gjennomsnittlige tilnæringsratioen  $\frac{L}{L_0}$  plottet som funksjon av  $\alpha$  og  $t$  for utvalgte verdier av  $k$ . Merk at gitt en svitsjingtid  $t$  er ratioen en diskret funksjon av  $\alpha$ , og gitt en fast  $\alpha$  er den en kontinuerlig funksjon av  $t$ . For ulike verdier av  $k$  har vi dessuten illustrert de beste valgene av  $\alpha$ , ved å plote som funksjon av  $t$  de verdiene av  $\alpha$  som gir lavest tilnæringsratio.

Vi observerer at for lave svitsjingtider får vi best resultater for lave verdier av  $\alpha$ . For  $t = 1$  gir H-TOP-L( $\alpha$ ) lavest tilnæringsratio for verdier av  $\alpha$  fra 1 til 6. Når  $T$  øker, øker også  $\alpha$  raskt. For resten av verdiene av  $t$  får vi best tilnærming for store verdier av  $\alpha$  lik eller nær 30, som er den maksimale verdien av  $\alpha$ . Hvor raskt  $\alpha$  øker til dette nivået varierer noe for de forskjellige verdiene av  $k$ , men det skjer for alle før  $t = 30$ .

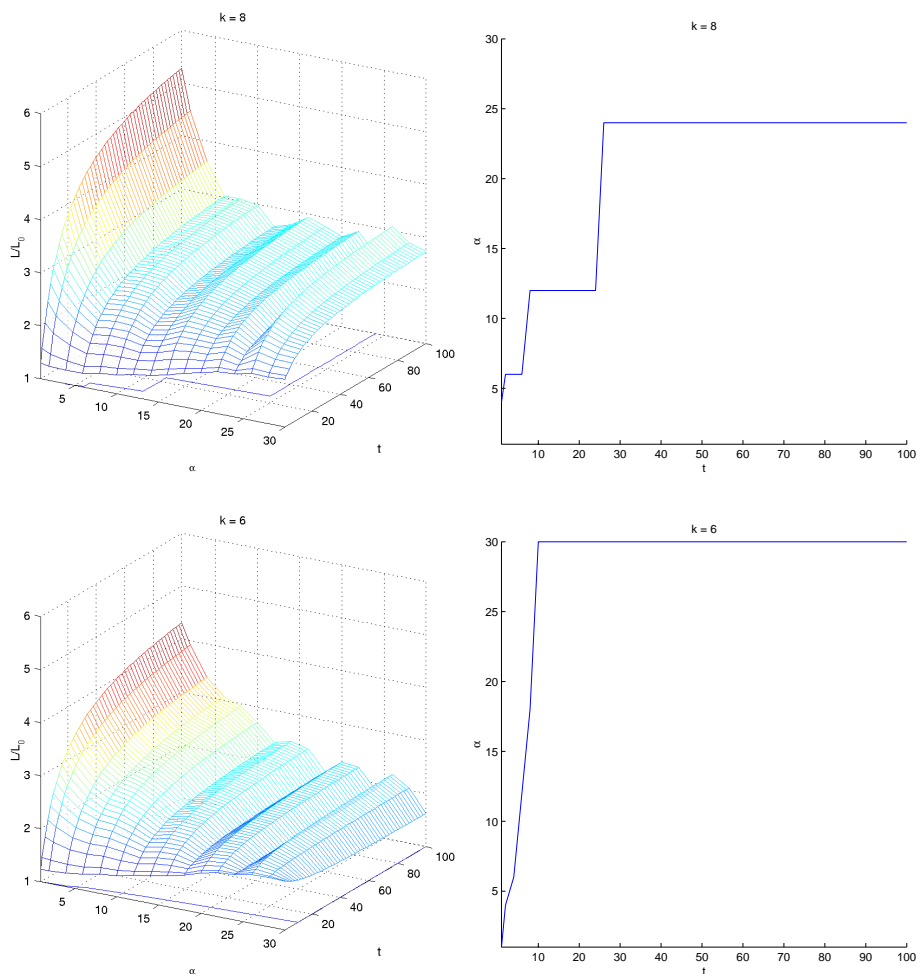


Figur 7.4: A-TOP-s og A-TOP-s-mod, gjennomsnittlig tilnærmingsratio  $\frac{L}{L_0}$  som som funksjon av antall antenner  $k$ .



Figur 7.5: Gjennomsnittlig tilnærmingsratio  $\frac{L}{L_0}$  for H-TOP-L( $\alpha$ ) som funksjon av  $\alpha$  og  $t$ .  $\alpha$  som gir beste ratio, som funksjon av  $t$ .  $k = 10$ .

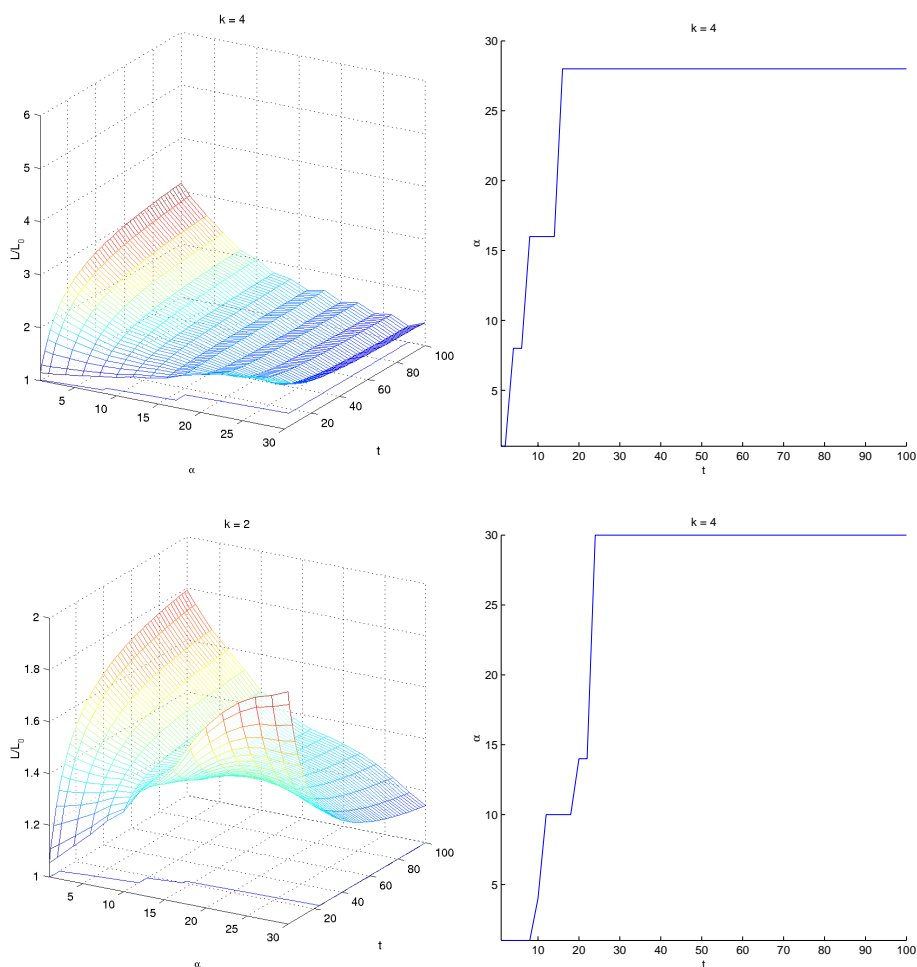




Figur 7.6: Gjennomsnittlig tilnærmingsratio  $\frac{L}{L_0}$  for H-TOP-L( $\alpha$ ) som funksjon av  $\alpha$  og  $t$ .  $\alpha$  som gir beste ratio, som funksjon av  $t$ .  $k = 8$  og  $k = 6$ .

For  $k$  lik 6, 4 og 2 legger vi merke til at den beste tilnærmingsratioen fås for  $\alpha = 1$ . Dette vil tilsvare bruk av A-TOP-T direkte på  $D$ .

Når vi i H-TOP-L( $\alpha$ ) konstruerer matrisen  $D_\alpha$ , sikrer vi at vi får maksimalt  $\frac{T_0(D)}{\alpha}$   $k$ -svitsjingmatriser. Ved å runde av de opprinnelige elementene i  $D$  oppover til nærmeste multiplum av  $\alpha$ , introduserer vi samtidig ekstra trafikk i systemet. Dette vil medføre en økning i overføringstiden  $T$ . For lave svitsjingtider vil betydningen av antall svitsjinger  $s$  være av mindre betydning enn  $T$  i den vektete summen  $L = T + st$ . Her vil vi dermed få best tilnærming for lave verdier av  $\alpha$ , som ikke gir en spesielt god



Figur 7.7: Gjennomsnittlig tilnærmingsratio  $\frac{L}{L_0}$  for H-TOP-L( $\alpha$ ) som funksjon av  $\alpha$  og  $t$ .  $\alpha$  som gir beste ratio, som funksjon av  $t$ .  $k = 4$  og  $k = 2$ .

øvre skranke for antall svitsjinger, men som derimot ikke gir mye ekstra trafikk i form av avrundinger. Med økende  $t$  vil antallet svitsjinger få økt vekt i  $L$ , og det nå fort et punkt der den beste tilnærmingen oppnås for høyere verdier av  $\alpha$ . Her ser vi at vi får lavest tilnærmingsratio ved å benytte samme svært høye verdi av  $\alpha$  for resten av verdiene av  $t$ . Forskjellen i økt trafikk ved bruk av en stor eller mindre stor  $\alpha$  teller lite i den vektete summen. Det gir dermed best uttelling å benytte en av de største tilgjengelige verdien av  $\alpha$ .

Ved tilnærming av SS/TDMA trafikkoptimeringsproblemet TOP-T med heuristikken H-TOP-L( $\alpha$ ) ser vi at det er viktig med valg av riktig  $\alpha$ . Ved store verdier av  $t$  vil det for alle  $k$  sannsynligvis være et trygt valg å be-

nytte en  $\alpha$  i nærheten av den maksimale verdien i trafikkmatrisen. For små verdier vil det beste valget av  $\alpha$  være mer avhengig av  $k$ , uten at vi i denne oppgaven på bakgrunn av de empiriske dataene kan trekke en entydig slutning om hvilket valg av  $\alpha$  som vil gi best tilnærming.

Beste valg av $\alpha$					
	$k$				
$t$	2	4	6	8	10
1	1	1	1	4	6
100	30	28	30	24	30

Tabell 7.3:  $\alpha$  som gir minst gjennomsnittlig tilnærmingsratio for H-TOP-L( $\alpha$ ).

#### 7.3.4 Beste heuristikk og tilnærmingsratio

Det finnes flere mulige kriterier for valg av den beste heuristikken. I denne empiriske analysen lar vi gjennomsnittlig tilnærmingsratio være avgjørende. Den beste heuristikken for SS/TDMA trafikkoptimeringsproblemet TOP-L er den som, gitt et antall antenner  $k$  og en svitsjingtid  $t$ , gir lavest gjennomsnittlig ratio  $\frac{L}{L_0}$  for ulike trafikkbehov som oppstår i satellittsystemet.

Tallmaterialet som ligger til grunn for valget er de 200 tilfeldig genererte trafikkmatrisene med verdier mellom 0 og 30. For disse er det konstruert TDMA skjemaer for systemer med antall antenner  $k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$  og svitsjingtider  $t$  fra 1 til 100. Det er benyttet heuristikkene A-TOP-s, A-TOP-s-mod, A-TOP-T og H-TOP-L( $\alpha$ ) som beskrevet i kapittel 6. H-TOP-L( $\alpha$ ) er testet for verdier av  $\alpha$  fra 1 til 30.

Figur 7.8 viser ytelsen av de forskjellige heuristikkene i form av gjennomsnittlig tilnærmingsratio  $\frac{L}{L_0}$  som funksjon av svitjingtiden  $t$ . Vi har plottet for antall antenner  $k$  lik 10, 8, 6, 4 og 2. For H-TOP-L( $\alpha$ ) har vi valgt to verdier av  $\alpha$ , som gir best ratio for henholdsvis  $t = 1$  og  $t = 100$ .

Vi ser klart at A-TOP-s-mod gir lavest tilnærmingsratio for alle andre enn de aller laveste svitsjingtiden  $t$ . Den umodifiserte algoritmen A-TOP-s kommer noe dårligere ut, og som tredje beste algoritme for store verdier av  $t$  finner vi H-TOP-L( $\alpha$ ) (da med valg av  $\alpha$  som gir best ratio for  $t = 100$ ). For  $t = 1$  gir A-TOP-T den laveste tilnærmingsratioen.

Avhengig av svitsjingtid  $t$  er dermed A-TOP-s-mod og H-TOP-L( $\alpha$ ) de bes-

te heuristikkene for TOP-L.

Vi ser også at for alle verdier av  $k$  og  $t$  får vi for den beste heuristikken en tilnærmingratio

$$\frac{L}{L_0} < 1.5$$

### 7.3.5 Kjøretider

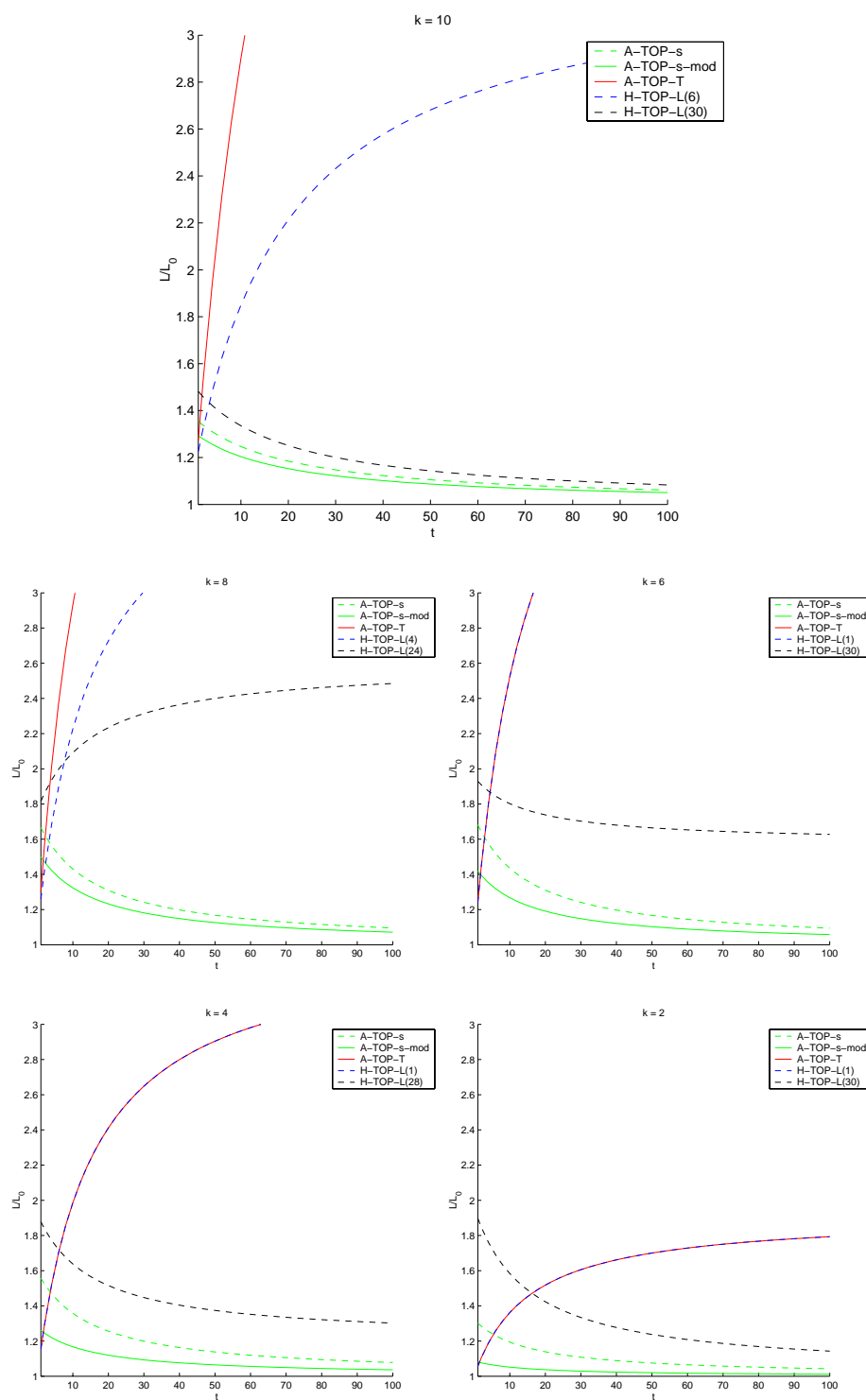
Som forklart i delkapittel 7.1 gjør valg av språk og matchingalgoritme at det ikke kan trekkes entydige slutninger om heuristikkenes reelle hastigheter fra implementasjonene i denne oppgaven. Vi skal likevel registrere noen kjøretider, for å få et inntrykk av forholdet algoritmene imellom.

På en  $10 \times 10$  trafikkmatrise med tilfeldig genererte elementer av verdi mellom 0 og 30 har vi kjørt algoritmene A-TOP-s, A-TOP-s-mod, A-TOP-T og H-TOP-L(2). For å måle tidsforbruket benyttes MATLAB-funksjonen *profile* ([21]). Denne har en presisjon på 0.010 sekunder. For at ikke presisjonen skal påvirke resultatene kjøres derfor hver algoritme 10 ganger på den samme trafikkmatrisen. Hver algoritme er kjørt for  $k = 5$  og  $k = 10$ . Beregningene er gjort i MATLAB versjon 6, Linux Redhat 7.3 operativsystem, på en maskin med Intel(R) Pentium(R) 4 2.0GHz prosessor.

Som vi ser av figur 7.4 bruker alle algoritmene som forventet lenger tid for  $k = 10$  enn for  $k = 5$ . Små verdier av  $k$  gir flere k-svisjningsmatriser og dermed flere iterasjoner av algoritmene. Vi kan også observere at A-TOP-s-mod bruker noe mer tid enn A-TOP-s. Dette skyldes først og fremst oppgavens implementasjon av kantordningen, som medfører økt antall matchingalgoritmer. A-TOP-T og H-TOP-L(2) bruker markant mer tid enn de andre algoritmene for  $k = 5$ . Disse minimerer i utgangspunktet overføringstiden  $T$ , og vi risikerer dermed TDMA skjemaer med noe flere svitsjinger og dermed flere iterasjoner i algoritmen.

Algoritme	$k$	Tid
A-TOP-s	5	0.68s
	10	0.18s
A-TOP-s-mod	5	1.68s
	10	0.76s
A-TOP-T	5	5.92s
	10	0.77s
H-TOP-L(2)	5	5.48s
	10	0.73s

Tabell 7.4: Kjøretider for algoritmer 10 ganger på  $10 \times 10$  trafikkmatrise.



Figur 7.8: Sammenlikning av heuristikker.

## Kapittel 8

# Oppsummering og konklusjon

I dette kapitlet oppsummerer vi arbeidet gjort i oppgaven. Vi gir også en konklusjon av funnene fra analysen av problem og algoritmer. Til slutt skisseres det hvordan det påbegynte arbeidet i oppgaven kan utvides.

### 8.1 Oppsummering

Vi har i oppgaven analysert SS/TDMA trafikkoptimeringsproblemet TOP-L, og kategorisert delproblemene og spesialtilfellene som behandles i litteraturen. For alle kategorier har vi så presentert resultater som gir nedre skranker for antall svitsjinger, overføringstid og total varighet av et TDMA skjema.

For delproblemene som minimerer antall svitsjinger og overføringstid har vi presentert bevis for at den nedre skranken kan oppnås, og at den optimale løsningen kan finnes i polynom tid. Algoritmene A-TOP-s og A-TOP-T som konstruerer TDMA skjemaer med henholdsvis minimum antall svitsjinger og overføringstid, er nøye forklart og implementert i MATLAB.

Vi har ikke forsøkt å finne en polynom algoritme som minimerer den totale varigheten av det generelle optimeringsproblemet. I spesialtilfellet TOP-L-k der antall antenner er lik den minimale antall sendere eller mottakere, har vi vist at problemet er *NP*-komplett. Vi har ikke klart å bevise det tilsvarende for andre antall antenner, men vi antar at også det generelle problemet er *NP*-komplett.

Det finnes dog klasser av problemet som lar seg løse optimalt i polynom tid. For spesialtilfellet TOP-L-k inntreffer dette for trafikkbehov gitt ved uvarierende matriser. For disse trafikkmatrisene vil den polynome algo-

ritmen A-TOP-L-u gi en optimal løsning av TOP-L-k. Dette resultatet har vi i denne oppgaven generalisert til å også omfatte problemet TOP-L. Vi har definert de k-uvarierende matrisene, som er en subklasse av de uvarierende. Det generelle problemet TOP-L med antall antenner  $k$  lar seg løse optimalt i polynom tid dersom trafikkmatrisen er k-uvarierende. Vi har vist at algoritmen A-TOP-L-u kan benyttes også i dette tilfellet. For å avgjøre i hvilken grad dette er praktisk anvendbare resultater, har vi analysert den prosentvis forekomsten av uvarierende og k-uvarierende matriser ved tilfeldig generering av matriser.

Grunnet optimeringsproblemets  $NP$ -komplekthet har vi presentert heuristikker for å finne gode tilnærminger til en optimal løsning. Vi har anvendt algoritmene A-TOP-s og A-TOP-T som heuristikker for TOP-L, og modifisert den første til A-TOP-s-mod for å gi en antatt bedre tilnærming.

Vi har også presentert en tilnærmingsalgoritme med bevisbart god tilnærming for spesialtilfellet TOP-L, H-TOP-L( $\alpha$ ). For det generelle tilfellet har vi ikke kunnet gi en tilsvarende garanti for tilnærmingsratioen. Vi velger derfor å betrakte H-TOP-L( $\alpha$ ) som heuristikk i tilfellene der antallet antenner er ekte mindre enn det minimale antallet.

Avslutningsvis har vi foretatt en empirisk analyse av alle heuristikkene, for å estimere hvor gode tilnærminger de gir i gjennomsnittlige tilfeller av problemet.

## 8.2 Konklusjon

Det viktigste teoretiske funnet i oppgaven er de k-uvarierende matrisene, som muliggjør en optimal løsning av det generelle SS/TDMA trafikkoptimeringsproblemet TOP-L i polynom tid. Som tallene i delkapittel 7.2 viser, er de prosentvise forekomstene av uvarierende og k-uvarierende tilfeldig genererte trafikkmatriser svært små. Det kan tenkes at slike trafikkmatriser oppstår oftere i reelle satellittsystemer, men vi må anta at dette resultatet først og fremst er av teoretisk verdi.

I den empiriske analysen av heuristikkens ytelse har vi funnet at det er A-TOP-s-mod og H-TOP-L( $\alpha$ ) som gir de beste gjennomsnittlige tilnærminger til optimal løsning av TOP-L. Gitt korte svitsjingtider gir H-TOP-L( $\alpha$ ) den beste tilnærmingen, mens for lengre svitsjingtider er A-TOP-s-mod best.

Ved valg av den beste heuristikken, får vi for alle testede svitsjingtider



og antall antenner gode tilnærminger til optimal løsning. De beste gjennomsnittlige tilnærmingsratioene er alle mindre enn 1.5. Til sammenlikning garanterer den beste tilnærmingsalgoritmen en tilnærmingsratio ekte mindre enn 2 for alle svitsjingtider i spesialtilfellet TOP-L-k.

### 8.3 Videre arbeid

Det gjenstår ennå mye i den teoretiske analysen av SS/TDMA trafikkoptimeringsproblemet. Aktuelt videre arbeid vil i første rekke innebære generalisering av kompleksitetsresultatene for TOP-L-k til det generelle problemet TOP-L. Vi har ennå ikke bevist at TOP-L er *NP*-komplett for alle antall antenner. Skulle vi ikke klare å vise dette, bør det oppmuntre oss til å lete etter algoritmer som gir optimale løsninger for enkelte eller flere antall antenner. Hvis det viser seg at TOP-L er *NP*-komplett ønsker vi å finne en nedre skranke for hvor gode tilnærminger vi kan garantere med polynome tilnærmingsalgoritmer. Det vil også være aktuelt å undersøke om  $H\text{-TOP-L}(\alpha)$  gir en øvre skranke for tilnærmingsratioen av TOP-L.

Både for TOP-L-k og for et eventuelt *NP*-komplett problem TOP-L vil vi i videre arbeid ønske å krympe avstanden mellom den nedre skranken for polynom tilnærming og den beste garanterte tilnærmingen. Dette kan skje gjennom leting etter både større nedre skranker og bedre tilnærmingsalgoritmer.

I den empiriske analysen av heuristikkene i oppgaven gjenstår det også arbeid. Vi har i oppgaven kun studert de gjennomsnittlige tilnærmingsratioene. En naturlig utvidelse vil være å også vurdere de dårligste tilnærmingsratioene for store antall tilfeldig genererte trafikkmatriser. En slik analyse kan gi estimerer på verste-tilfelle ytelsen av heuristikkene, og kan således tjene til å oppmuntre eller fraråde leting etter større nedre skranke for polynom tilnærming.



# Tillegg A

## Programkode

Programkoden er vedlagt i sin helhet, strukturert etter hvilke algoritmer den implementerer. Funksjonsfiler som benyttes i flere algoritmer er kun tatt med én gang, under den første algoritmen der de benyttes.

### A.1 Bestemme (k-)uvarierende matriser

#### A.1.1 unvarying.m

```
%unvarying
%
%u = unvarying(D,m,n)
%
%Returnerer den minste k som D er k-uvarierende for, hvis u = min(m,n), da
%er matrisen uvarierende. Hvis u = 0 er ikke matrisen uvarierende.

function u = unvarying(D,m,n)

u = min(m,n);

%Finner de forskjellige verdiene som finnes i D og hvor mange elementer det
%finnes av hvert element

Dtmp = D;

values = max(max(Dtmp));
numval = max(size(find(Dtmp == values)));

Dtmp = Dtmp - (Dtmp == values)*values;

l = 1;

while (max(max(Dtmp)) ~= 0)

    l = l + 1;
    values = cat(2,values,max(max(Dtmp)));
    numval = cat(2,numval,max(size(find(Dtmp == values(1,l)))));

    Dtmp = Dtmp - (Dtmp == values(1,l))*values(1,l);

end
```

```

%Det største antallet positive elementer i en rad eller linje
maxnum = max([max(sum(D'>0)) max(sum(D>0))]);

%Går gjennom radene med flest positive elementer og finner det minste
%antallet av en verdi w.
min_numvalues = ones(size(values))*max([m n]);

for i = 1:m
    if sum(D(i,:)>0) == maxnum
        for j = 1:l
            min_numvalues(j) = min([min_numvalues(j) sum(D(i,:) == values(j))]);
        end
    end
end

%Tilsvarende for kolonnene
for i = 1:n
    if sum(D(:,i)>0) == maxnum
        for j = 1:l
            min_numvalues(j) = min([min_numvalues(j) sum(D(:,i) == values(j))]);
        end
    end
end

%Sjekker om det finnes en rad med flere elementer av en verdi enn antall
%elementer av verdi i en av radene med flest positive elementer. I så fall
%er ikke D uvarierende.

i = 0;
while (u ~= 0) & i < m
    i = i + 1;
    for j = 1:l
        if (sum(D(i,:) == values(j)) > min_numvalues(j))
            u = 0; %Matrisen er ikke uvarierende
            break;
        end
    end
end

%Tilsvarende for kolonnene
i = 0;
while (u ~= 0) & i < n
    i = i + 1;
    for j = 1:l
        if (sum(D(:,i) == values(j)) > min_numvalues(j))
            u = 0; %Matrisen er ikke uvarierende
            break;
        end
    end
end

%Dersom D er uvarierende sjekker vi om den også er k-uvarierende for
%en  $k < \min(m,n)$ .

if u ~= 0
    k = min(m,n);
    while (u == k) & (k > 1)

```

```

k = k - 1;
u = k;

j = 1;
while (u == k) & (j <= 1)
    if (ceil(numval(j)/k) > min_numvalues(j))
        u = k+1; %D er ikke k-uvarierende
    end
    j = j + 1;
end

end
end

```

## A.2 Algoritme A-TOP-s

### A.2.1 min\_s.m

```

% min_s
%
% s = min_s(D,m,n,k) Finner minste antall svtjinger i en TDMA frame
% D er mxn trafikkmatris, og k er antall transpondere

function s = min_s(D,m,n,k)

%Finn maksimale antall positive elementer i en rad eller kolonne

v = 0;

for i = 1:m
    v_tmp = size(find(D(i,:)));
    if v_tmp(2) > v
        v = v_tmp(2);
        if v == max([m n])
            break;
        end
    end
end

if v ~= max([m n])
    for i = 1:n
        v_tmp = size(find(D(:,i)));
        if v_tmp(1) > v
            v = v_tmp(1);
            if v == max([m n])
                break;
            end
        end
    end
end

if (k < max([m n]))
    rho = max(size(find(D))); %Antall positive elementer i D

```

```

    s = max(v,ceil(rho/k));
else
    s = v;
end

```

### A.2.2 min\_T.m

```

% min_T
%
% T = min_T(D,m,n,k) Finner minimum overføringstid i en TDMA frame
% D er mxn trafikkmatris, og k er antall transpondere

function T = min_T(D,m,n,k)

%Finn maksimal rad – eller kolonnesum

u = max([ max(sum(D)) max(sum(D'))]);

if ( k < max([m n]) )
    sigma = sum(sum(D)); %Summen av elementene i D
    T = max(u,ceil(sigma/k));
else
    T = u;
end

```

### A.2.3 dummy\_v.m

```

%dummy_v
%
%function v = dummy_v(r,d,l)
%
%Finner graden til en node, representert ved en vektor r der de l første
%elementene er kanter med vekt lik elementets verdi.
%Vektoren d er en vektor med dummykanter, der verdien av elementet er antall
%kanter med vekt lik 0.

function v = dummy_v(r,d,l)

t = max(size(r));

v = max(size(find(r(1:l) - d(1:l)))) + sum(d);

```

### A.2.4 add\_dummy\_A\_TOP\_s.m

```

%add_dummy_A_TOP_s
%
%[Dd,dummy] = add_dummy_A_TOP_s(D,k,s0) konstruerer matrise for graf med dummy-kanter
%som i algoritme A-TOP-s. D er trafikkmatrisen, k antall antenner og s0 det
%minimale antall svitsjinger i en TDMA frame gitt D og k.

function [Dd,dummy] = add_dummy_A_TOP_s(D,k,s0)

```

```

[m,n] = size(D);

%Steg 2
%Addisjon av ekstra noder

N = n + m - k;

Dd = zeros(N,N);
dummy = zeros(N,N); %Gir lokasjonen for dummy-elementene med vekt lik 0

Dd(1:m,1:n) = D;

%Steg 3
%Addisjon av ekstra kanter

%Steg 3.1
for i = m+1:N
    for j = 1:n
        while ( sum(Dd(i,:)) < s0 ) & ( dummy_v(Dd(:,j),dummy(:,j),m) < s0 )
            Dd(i,j) = Dd(i,j) + 1;
            dummy(i,j) = dummy(i,j) + 1;
        end
    end
end

%Steg 3.2
for j = n+1:N
    for i = 1:m
        while ( sum(Dd(:,j)) < s0 ) & ( dummy_v(Dd(i,:),dummy(i,:),n) < s0 )
            Dd(i,j) = Dd(i,j) + 1;
            dummy(i,j) = dummy(i,j) + 1;
        end
    end
end

%Steg 3.3
for i = 1:m
    for j = 1:n
        while ( dummy_v(Dd(i,:),dummy(i,:),n) < s0 ) & ( dummy_v(Dd(:,j),dummy(:,j),m) < s0 )
            Dd(i,j) = Dd(i,j) + 1;
            dummy(i,j) = dummy(i,j) + 1;
        end
    end
end

```

### A.2.5 frame\_A\_TOP\_s.m

```

%frame_A_TOP_s
%
%[FRAME,EXT,s0,T] = frame_A_TOP_s(D,m,n,k)
%Konstruerer TDMA frame med minimalt antall svitsjinger gitt mxn
%trafikkmatrise D og k antenner. Returnerer antall svitsjinger,
%overføringstid, k-svitsjingmatrisene i TDMA framen og matrisene med
%dummyelementer brukt i løpet av kjøringen.

function [FRAME,EXT,s0,T] = frame_A_TOP_s(D,m,n,k)

```

```

%Steg 1

s0 = min_s(D,m,n,k);

T = 0; %Overføringstiden T

FRAME = zeros(m,n,s0); %Holder de s0 k-svitjingmatrisene

%Steg 2 og 3

[Dd,dummy] = add_dummy_A_TOP_s(D,k,s0);

EXT = zeros(m+n-k,m+n-k,s0); %Holder de midlertidige utvidede trafikkmatrisene

%Steg 4 og 5

for s = 1:s0

    EXT(:,s) = Dd;
    [FRAME(:,s),Dd,dummy] = switching_A_TOP_s(Dd,dummy,m,n);
    T = T + max(max(FRAME(:,s))); %Oppdaterer overføringstiden T

end

```

### A.2.6 s0T\_A\_TOP\_s.m

```

%s0T_A_TOP_s
%
%[s0,T] = s0T_A_TOP_s(D,m,n,k) Konstruerer TDMA frame med minimalt antall
%svitsjinger gitt mxn trafikkmatrise D og k antenner. Returnerer kun antall
%svitjinger og overføringstid.

function [s0,T] = s0T_A_TOP_s(D,m,n,k)

%Steg 1

s0 = min_s(D,m,n,k);

T = 0; %Overføringstiden T

S = zeros(m,n); %Holder k-svitjingmatrisen

%Steg 2 og 3

[Dd,dummy] = add_dummy_A_TOP_s(D,k,s0);

for s = 1:s0

    [S,Dd,dummy] = switching_A_TOP_s(Dd,dummy,m,n);
    T = T + max(max(S)); %Oppdaterer overføringstiden T

end

```

### A.2.7 switching\_A\_TOP\_s.m

```

%switching_A_TOP_s
%
%[S,Dd2,dummy2] = switching_A_TOP_s(Dd,dummy,m,n)

```



*%Finner en maksimum matching i matrise Dd med dummy-elementer, og returnerer  
%en k-svitsjingmatrise S og Dd til bruk i neste iterasjon av algoritme A-TOP-s.*

```
function [S,Dd2,dummy2] = switching_A_TOP_s(Dd,dummy,m,n)

dummy2 = dummy;
N = max(size(Dd));
M1 = zeros(N,N); %Skal holde matching
M2 = zeros(N,N); %Skal holde matching minus dummy-kanter

p = dmperm(Dd); % Dulmage-Mendelsohn decomposition

for j = 1 : N
    if p(j) ~= 0

        if dummy(p(j),j) > 0
            M1(p(j),j) = 1;
            dummy2(p(j),j) = dummy(p(j),j) - 1;
        else
            M1(p(j),j) = Dd(p(j),j);
            M2(p(j),j) = Dd(p(j),j);
        end

    end
end

Dd2 = Dd - M1;

S = M2(1:m,1:n);
```

## A.3 Algoritme A-TOP-s-mod

### A.3.1 mod\_switchingATOPs.m

*%mod\_switching\_A\_TOP\_s  
%  
%[S,Dd2,dummy2] = mod\_switchingATOPs(Dd,dummy,m,n)  
%Finner en maksimum matching i matrise Dd med dummy-elementer, og returnerer  
%en k-svitsjingmatrise S og Dd til bruk i neste iterasjon av algoritme A-TOP-s-mod.*

```
function [S,Dd2,dummy2] = mod_switchingATOPs(Dd,dummy,m,n)

dummy2 = dummy;
N = max(size(Dd));
M1 = zeros(N,N); %Skal holde matching
M2 = zeros(N,N); %Skal holde matching minus dummy-kanter

Dtmp = Dd(1:m,1:n);

maxd = max(max(Dtmp));
Delements = (Dtmp >= maxd)*maxd;
Dtmp = Dtmp - Delements;
e = 1;

while (max(max(Dtmp)) > 0)

    e = e + 1;
```

```

    maxd = max(max(Dtmp));
    Delements = cat(3,Delements,(Dtmp >= maxd)*maxd) ;
    Dtmp = Dtmp - Delements(:,e);

end

Dtest = Dd(1:m,1:n);
for s = 1:e
    Dtest = Dtest - Delements(:,s);
end

e = 1;
Ddtmp = Dd;
Ddtmp(1:m,1:n) = Delements(:,e);

p = dmperm(Ddtmp); %Dulmage–Mendelsohn decomposition
p_size = size(find(p));

while (p_size(2) < N)

    e = e + 1;
    Ddtmp(1:m,1:n) = Ddtmp(1:m,1:n) + Delements(:,e);

    p = dmperm(Ddtmp);
    p_size = size(find(p));

end

for j = 1 : N
    if p(j) ~= 0

        if dummy(p(j),j) > 0
            M1(p(j),j) = 1;
            dummy2(p(j),j) = dummy(p(j),j) - 1;
        else
            M1(p(j),j) = Dd(p(j),j);
            M2(p(j),j) = Dd(p(j),j);
        end
    end
end

Dd2 = Dd - M1;

S = M2(1:m,1:n);

```

## A.4 Algorithme A-TOP-T

### A.4.1 add\_dummy\_ATOPT.m

```

%add_dummy_ATOPT
%
%[D2,dummy] = add_dummy_ATOPT(D,m,n,k,R)

```

```

%
%Legger trafikk til den mxn trafikkmatrisen D, inntil summen av elementene
%er lik kR.

function [D2,dummy] = add_dummy_ATOPT(D,m,n,k,R)

N = m + n - k

dummy = zeros(m,n); %Gir lokasjonen for dummy-elementene fra steg 2

D2 = D;

%Steg 2
%Addisjon av dummytrafikk

for i = 1:m
    for j = 1:n
        sigma = sum(sum(D2));

        if sigma < k*R

            q = min((R - sum(D2(i,:))) (R - sum(D2(:,j))) (k*R - sigma));

            D2(i,j) = D2(i,j) + q;
            dummy(i,j) = dummy(i,j) + q;

        else
            break;
        end

    end

    if sigma == k*R
        break;
    end
end

end

```

#### A.4.2 extend\_ATOPT.m

```

%extend_ATOPT
%
%[Dd,N] = extend_ATOPT(D,m,n,k,R)
%
%Utvider den mxn trafikkmatrisen D med ekstra n-k rader og m-k kolonner, og
%og legger til ekstra trafikk inntil alle linjesummer er lik R.

function [Dd,N] = extend_ATOPT(D,m,n,k,R)

N = n + m - k;

Dd = zeros(N,N);

Dd(1:m,1:n) = D;

%Steg 3

```

```

for i = m+1:N
    for j = 1:n

        q = R - max([sum(Dd(i,:)) sum(Dd(:,j))]);

        Dd(i,j) = Dd(i,j) + q;

    end
end

for j = n+1:N
    for i = 1:m

        q = R - max([sum(Dd(i,:)) sum(Dd(:,j))]);

        Dd(i,j) = Dd(i,j) + q;

    end
end

```

### A.4.3 switching\_ATOPT.m

```

%switching_ATOPT
%
%[S,D3,dummy2,R2] = switching_ATOPT(Dd,D2,dummy,m,n,N,k,R)
%
%Finner en maksimum matching i matrisen Dd med dummy-elementer, og
%returnerer en k-svisjningmatrise S og trafikkmatrise D3 til bruk i neste
%iterasjon av algoritme A-TOP-T.

function [S,D3,dummy2,R2] = switching_ATOPT(Dd,D2,dummy,m,n,N,k,R)

M = zeros(N,N);

p = dmperm(Dd); % Dulmage-Mendelsohn decomposition

for j = 1 : N
    if p(j) ~= 0

        M(p(j),j) = Dd(p(j),j);

    end
end

%Finne største linjesum i D2 (med dummy-trafikk) som ikke er dekket av matchingen

MD = M(1:m,1:n);

Lpos_col = find(sum(MD));
Lpos_row = find(sum(MD'));

Dtmp = D2;
Dtmp(:,Lpos_col) = 0;

L = max(sum(Dtmp)); %Største kolonnesum i D2 ikke dekket av matchingen.

Dtmp = D2;
Dtmp(Lpos_row,:) = 0;

```

```

L = max([L max(sum(Dtmp'))]); %Største linjesum i D ikke dekket av matchingen

if R == L
    b = min([MD(find(MD))]);
else
    b = min([(R - L) MD(find(MD))]);
end

M = (M > 0)*b;

dummy2 = zeros(N,N);
dummy2(1:m,1:n) = dummy;

D3 = D2 - M(1:m,1:n);

R2 = R - b;

%Fjerner dummy-trafikken

for j = 1:N
    if p(j) ~= 0
        if dummy2(p(j),j) ~= 0
            if (b >= dummy2(p(j),j) )
                M(p(j),j) = b - dummy2(p(j),j);
                dummy2(p(j),j) = 0;
            else
                M(p(j),j) = 0;
                dummy2(p(j),j) = dummy2(p(j),j) - b;
            end
        end
    end
end

dummy2 = dummy2(1:m,1:n);

S = M(1:m,1:n); %k-svisingmatrise med dummy-trafikk trukket fra

```

#### A.4.4 frame\_ATOPT.m

```

%frame_ATOPT
%
%[FRAME,EXT,s,T0] = frame_ATOPT(D,m,n,k)
%
%Konstruerer TDMA frame med minimal overføringstid gitt mxn trafikkmatrise D
%og k-antennene. Returnerer overføringstid, antall svisjinger, de utvidede
%matrisene konstruert i løpet av kjøringen og k-svisingmatrisene.

function [FRAME,EXT,s,T0] = frame_ATOPT(D,m,n,k)

%Steg 1

T0 = min_T(D,m,n,k);

R = T0;

```

```

s = 0;

FRAME = zeros(m,n); %Holder svitjingmatrisene
EXT = zeros(m,n); %Holder de utvidede trafikkmatrisene

%Steg 2 og 3
[D2,dummy] = add_dummy_ATOPT(D,m,n,k,R);

%Steg 4 og 5
[Dd,N] = extend_ATOPT(D2,m,n,k,R);
[FRAME,D2,dummy,R] = switching_ATOPT(Dd,D2,dummy,m,n,N,k,R);
EXT = Dd;

s = s + 1;

while ( R > 0 )

    [Dd,N] = extend_ATOPT(D2,m,n,k,R);
    [S,D2,dummy,R] = switching_ATOPT(Dd,D2,dummy,m,n,N,k,R);

    FRAME = cat(3,FRAME,S);
    EXT = cat(3,EXT,Dd);

    s = s + 1;

end

```

#### A.4.5 T0s\_ATOPT.m

```

%T0s_ATOPT
%
%[T0,s] = T0s_ATOPT(D,m,n,k)
%
%Konstruerer TDMA frame med minimal overføringstid gitt mxn trafikkmatrise D
%og k-antennner. Returnerer kun overføringstid og antall svitsjinger.

function [T0,s] = T0s_ATOPT(D,m,n,k)

%Steg 1

T0 = min_T(D,m,n,k);

R = T0;

s = 0;

%Steg 2 og 3
[D2,dummy] = add_dummy_ATOPT(D,m,n,k,R);

%Steg 4 og 5
[Dd,N] = extend_ATOPT(D2,m,n,k,R);
[FRAME,D2,dummy,R] = switching_ATOPT(Dd,D2,dummy,m,n,N,k,R);

s = s + 1;

while ( R > 0 )

```

```

[Dd,N] = extend_ATOPT(D2,m,n,k,R);
[S,D2,dummy,R] = switching_ATOPT(Dd,D2,dummy,m,n,N,k,R);

s = s + 1;

end

```

## A.5 Algoritme H-TOP-L( $\alpha$ )

### A.5.1 frame\_HTOPLalpha.m

```

%frame_HTOPLalpha
%
%[FRAME,s,T] = frame_HTOPLalpha(D,m,n,k,alpha)
%
%%Konstruerer TDMA skjema med heuristikk H-TOP-L(alpha) gitt mxn
%trafikkmatrix D og k antenner. Returnerer skjema, antall svitsjinger og
%overføringstid.

function [FRAME,s,T] = frame_HTOPLalpha(D,m,n,k,alpha)

Dalpha = ceil(D/alpha)*alpha;

[FRAME,EXT,s,T] = frame_ATOPT(Dalpha,m,n,k);

```

### A.5.2 sT\_HTOPLalpha.m

```

function [s,T] = sT_HTOPLalpha(D,m,n,k,alpha)

Dalpha = ceil(D/alpha)*alpha;

[T,s] = T0s_ATOPT(Dalpha,m,n,k);

```

## A.6 Simulering

### A.6.1 sim\_alg.m

```

%Test av tilnærmingsalgoritmer for TOP-L
%for tilfeldig genererte trafikkmatiser

clear all;

%Dimensjon på trafikkmatisene

n = 10;

%Svitjingtidene t

t = [1 2:2:100];
t_size = size(t);

```

```

%Verdiene av k

k = [1 2 3 4 5 6 7 8 9 10];
k_size = size(k);

%Verdiene av alpha
alpha = [1 2:2:30];
alpha_size = size(alpha);

%Max størrelse på element i trafikkmatrise

max_d = 30;

%Antall tilfeldige trafikkmatriser generert

num = 200;

%Resultatmatrise (n,k,t) med gjennomsnittlig (1) og dårligste(2)
%tilnærmingssratio L/L min (dimensjon 4), for
%A-TOP-s(1),A-TOP-s-mod(2),A-TOP-T(3),H-TOP-L(alpha)(4) algoritmene
%(dimensjon 5).

L = zeros(k_size(2),t_size(2),2,4,alpha_size(2));

%Midlertidig matrise til å holde resultat

L_tmp = zeros(num,k_size(2),t_size(2),4,alpha_size(2));

for p = 1:num

    p
    %Random ixi trafikkmatrise, verdier < max_d
    D = round(rand(n)*max_d);

    for l = 1:k_size(2)

        %Kjører algoritmer
        [s0,T] = s0T_A_TOP_s(D,n,n,k(l)); %A-TOP-s
        [s0,Tmod] = mod_s0T_ATOPs(D,n,n,k(l)); %A-TOP-s-mod
        [T0,s] = T0s_ATOPT(D,n,n,k(l)); %A-TOP-T

        for j = 1:t_size(2)
            %j
            L_tmp(p,l,j,1,1) = (T + t(j)*s0)/(T0 + t(j)*s0); %A-TOP-s
            L_tmp(p,l,j,2,1) = (Tmod + t(j)*s0)/(T0 + t(j)*s0); %A-TOP-s-mod
            L_tmp(p,l,j,3,1) = (T0 + t(j)*s)/(T0 + t(j)*s0); %A-TOP-T
        end

        %H-TOP-L(alpha)
        for a = 1:alpha_size(2)
            %a
            [s_alpha,T_alpha] = sT_HTOPLalpha(D,n,n,k(l),alpha(a));
            for j = 1:t_size(2)
                L_tmp(p,l,j,4,a) = (T_alpha + t(j)*s_alpha)/(T0 + t(j)*s0);
            end
        end
    end
end
end

```



end

for l = 1:k\_size(2)

for j = 1:t\_size(2)

%j

*%Regner gjennomsnittlig og dårligste ratio for*

*%A-TOP-s(f=1),A-TOP-s-mod(f=2) og A-TOP-T(f=3)*

for f = 1:3

L(l,j,1,f,1) = mean(L\_tmp(:,l,j,f,1)); *%Gjennomsnittelig tilnærmingsratio*

L(l,j,2,f,1) = max(L\_tmp(:,l,j,f,1)); *%Dårligste tilnærmingsratio*

end

*%Renger gjennomsnitt og dårligste ratio for H-TOP-L(alpha)*

for a = 1:alpha\_size(2)

L(l,j,1,4,a) = mean(L\_tmp(:,l,j,4,a)); *%Gjennomsnittelig ratio*

L(l,j,2,4,a) = max(L\_tmp(:,l,j,4,a)); *%Dårligste ratio*

end

end

end

save sim060503.mat



# Bibliografi

- [1] F. Afrati, T. Asladnidis, E. Bampis og I. Millis. Scheduling in switching networks with set-up delays. <http://hipercom.inria.fr/algotel2002/actes/54.ps.gz>.
- [2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela og M. Protasi. *Complexity and approximation: combinatorial optimization problems and their approximability properties*. Springer-Verlag, 1999.
- [3] A. Björner og R. P. Stanley. A combinatorial miscellany. <http://www.math.kth.se/bjorner/files/CUP.pdf>, 1998.
- [4] G. Bongiovani og Wong C.K. Coppersmith D. An optimum slot assignment algorithm for an ss/tdma system with variable number of transponders. *IEEE. Trans. Comm.*, (Com-29):721–726, 1981.
- [5] R. A. Brualdi. Some applications of doubly stochastic matrices. *Linear Algebra and its applications*, (107):77–100, 1988.
- [6] William J. Cook, William H. Cunningham, William R. Pulleyblank og Alexander Schrijver. *Combinatorial Optimization*. Wiley Interscience, 1998.
- [7] P. Crescenzi, D. Xiaotie og C.H. Papadimitriou. On approximating a scheduling problem. *Journal of combinatorial optimization*, 5:287 – 297, 2001.
- [8] Geir Dahl. An introduction to convexity, polyhedral theory and combinatorial optimization. *Kompendium* 67, 1997.
- [9] S. Even, A. Itai og A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5:691 – 703, 1976.
- [10] I.S. Gopal og C.K. Wong. Minimizing the number of switchings in an ss/tdma system. *IEEE. Trans. Comm.*, (Com-33):497–501, 1985.

- [11] Han Hoogeveen, Petra Schuurman og Gerhard J. Woeginger. Non-approximability results for scheduling problems with minsum criteria. *Lecture Notes in Computer Science*, 1412:353-??, 1998.
- [12] J.A. Hoogeveen, J.K. Lenstra og B. Veltman. Three, four, five, six, or the complexity of scheduling delays. *Operations research letters*, 16:129 - 137, 1994.
- [13] J.E. Hopcroft og R.M. Karp. An  $n^{\frac{5}{2}}$  algorithm for maximum matching in bipartite graphs. *SIAM J. Comput.*, 2:225 - 231, 1973.
- [14] E.L. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, 1976.
- [15] L. Lovasz og M.D. Plummer. *Matching Theory*. Academic Press, 1986.
- [16] G.L. Nemhauser og L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.
- [17] G. Nemhauser. The age of optimization: Solving large-scale real-world problems. *Operations Research*, 42(1):5-13, 1994.
- [18] A. Pothén og F. Chin-Ju. Computing the block triangular form of a sparse matrix. *ACM Transactions on Mathematical Software*, 16(4):303 - 324, 1990.
- [19] F Rendl. On the complexity of decomposing matrices arising in satellite communicationc. *Operations research letters*, 4, 1985.
- [20] M.A. Weiss. *Data structures and algorithms analysis in Java*. Addison-Wesley, 1999.
- [21] *Using MATLAB (Version 6)*. The MathWorks, Inc., 2000.